On Multi-Dimensional Privacy in Context-Aware Mobile Networks

THÈSE N^{O} 6066 (2014)

PRÉSENTÉE LE 7 MARS 2014

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE POUR LES COMMUNICATIONS INFORMATIQUES ET LEURS APPLICATIONS 1
PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Igor BILOGREVIC

acceptée sur proposition du jury:

Prof. P. Thiran, président du jury
Prof. J.-P. Hubaux, directeur de thèse
Prof. M. Grossglauser, rapporteur
Prof. M. Langheinrich, rapporteur
Dr S. Schnorf, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2014

Not everything that can be counted counts,
not everything that counts can be counted.
— William Bruce Cameron, 1963

To my family

Abstract

By the end of 2013, the number of internet-connected mobile devices is expected to exceed that of humans. Omnipresent and context-aware, mobile devices enable people to communicate and exchange data anytime and almost anywhere. The myriad of "digital footprints" that mobile devices leave can be used to infer a large amount of personal information about their owners. For instance, the IP address can be used to infer a coarse-grained location of the device, the temporary identifiers used in cellular networks can be used to track people's whereabouts and infer numerous personal details. Similarly, online social networks often force members to share some personal information with all other users or service providers, de facto exposing users to unwanted profiling by advertisement companies and other private and state agencies. At each layer of the network stack, there is some information that can be used to track and profile mobile users; it is therefore crucial to investigate the privacy challenges present at different layers and design privacy protection mechanisms that work across these layers.

In this thesis, we take a top-down approach on privacy in mobile networks by (i) studying the issues present in different network layers – the application, IP and link layers – and (ii) by proposing protection mechanisms and quantifying the extent of private information leakage. First, we look at the application layer, where we design protocols to protect users' personal data from third-party entities and other unauthorized users. In particular, we focus on two relevant problems: meeting scheduling and optimal meeting location determination. For these two problems, we propose and evaluate privacy-preserving protocols that are both practical and more efficient than the existing approaches. Second, we study the privacy challenges that arise in the network and link layers, by quantifying the exposure of social community information in a large on-campus experiment. In addition, we evaluate the effect of the reconstructed community information on the inference of social ties among the participants to the experiment. For the first time in the same experiment, we compare the reconstruction accuracy of a realistic eavesdropper, who has only access to packet headers exchanged among the mobile devices, with that of a malicious application or entity that has

access to the on-device data. Third, by taking a cross-layer approach, we design and evaluate a mobile social-networking application that enables users to share different kinds of personal information in a privacy-aware and inobtrusive way. In particular, we show how existing information-sharing policies are ineffective in correctly predicting users actual sharing behavior; then, based on a probabilistic decision-making framework, we demonstrate how machine learning can be used to automatically decide whether and how much to share – based on the users' context and past behavior. Our results indicate that the proposed machine-learning-based approach is more comprehensive and practical than existing automated solutions and, at the same time, it is more effective than fixed policy-based rules – all while requiring a minimal effort from the users.

Keywords: Mobile networks, privacy, meeting scheduling, encryption, social communities, decision-making, user-study, machine learning

Riassunto

Entro la fine del 2013, si prevede che il numero di dispositivi mobili connessi ad Internet sorpasserà il numero di persone. Onnipresenti e sensibili al contesto dell'utente, i dispositivi mobili permettono alle persone di comunicare e scambiare dati in qualsiasi momento ed in quasi ogni luogo. La miriade di "impronte digitali" che questi dispositivi lasciano possono essere usati per inferire una grande quantità di informazioni personali sui loro proprietari. Ad esempio, l'indirizzo IP può essere utilizzato per dedurre la posizione approssimativa del dispositivo, l'identificativo temporaneao impiegato nelle reti di telefonia cellulare può venir sfruttato per tener traccia degli spostamenti delle persone and per inferire numerosi dettagli personali. Analogamente, gli online social networks obbligano spesso i loro iscritti a condividere alcune informazioni personali con tutti gli altri membri, esponendo di fatto gli utenti ad analisi comportamentali condotte da compagnie pubblicitarie ed altre agenzie, sia private che pubbliche. Su ogni layer dello stack di rete, vi sono delle informazioni che possono venir utilizzate per tener traccia e fare profiling di persone; diventa quindi cruciale l'investigazione delle sfide nell'ambito della privacy su differenti layers, così come lo sviluppo di meccanismi di protezione della privacy che lavorino trasversalmente fra diversi layers.

Nella presente tesi, viene preso un'approccio *top-down* rispetto alla privacy nelle reti mobili. In primo luogo, viene presentato lo studio delle sfide presenti su diversi layers dello stack di rete – link, IP e applicativo. In secondo luogo, vengono proposti meccanismi di protezione e quantificata l'estensione della fuga di dati privati. In primo luogo, ci focalizziamo sul layer applicativo, dove sviluppiamo protocolli per proteggere i dati personali degli utenti da parti terze e utenti non autorizzati. In particolare, ci concentriamo su due problemi rilevanti: pianificazione di attività e determinazione di una località ottimale. Per questi due problemi, proponiamo e esaminiamo protocolli che preservano la privacy e sono sia pratici che più efficienti di soluzioni esistenti. In secondo luogo, proponiamo uno studio sulle sfide della privacy che si manifestano sui layers network e link, dove quantifichiamo l'estensione dell'esposizione di informazioni private riguardanti le communità

sociali delle persone in un'ampio esperimento condotto sul campus universitario. In aggiunta, esaminamo l'effetto della ricostruzione delle comunità sulla capacit di inferenza del tipo di relazioni sociali fra i partecipanti. Vengono comparati, per la prima volta nello stesso esperimento, l'accuratezza dell'inferenza effettuata da un origliatore esterno (il quale non ha accesso ai dati contenuti sul terminale mobile) con quella di un'applicazione maligna o entità che ha accesso ai dati presenti sul terminale. In terzo luogo, basandoci su un'approccio trasversale attraverso diversi layers, sviluppiamo ed esaminiamo un'applicazione sociale per terminali mobili che permette agli utenti di condividere informazioni personali in modo rispettoso della privacy e non intrusivo. In particolare, mostriamo come le regole di condivisione esistenti sono inefficienti nel predire correttamente il comportamento dell'utente quando condivide l'informazione; in seguito, basandoci su una struttura probabilistica decisionale, dimostriamo come il machine-learning può essere utilizzato per decidere in modo automatico se e a che dettaglio l'informazione verrà condivisa con altri – a seconda del contesto attuale e del comportamento nel passato. I nostri risultati indicano che il sistema proposto é più completo e pratico rispetto a sistemi esistenti ed é, allo stesso tempo, più efficace di sistemi basati su policies fisse – il tutto richiedendo uno sforzo minimale all'utente.

Parole chiave: Reti mobili, privacy, pianificazione di attività, crittografia, comunità sociali, presa di decisione, studi con persone, machine learning

Acknowledgements

This thesis has greatly benefited from the knowledge, expertise and input of several people to whom I am deeply and sincerely grateful. The four years that led to its completion have been a wonderful rollercoaster ride thanks to the many of you who made it so memorable! Before expressing my gratitude to the people whose names appear hereafter, I would like to sincerely thank all those who are not mentioned, including the students and interns I advised, the numerous colleagues at Nokia whom I had the pleasure to work with, to my dearest friends who shared the joys and tears with me, to our four wonderful members of the administrative staff and to people in our IT support team. To all of you, thank you.

First and foremost, I would like to thank my advisor Prof. Jean-Pierre Hubaux who provided me with guidance, experience and inspiration much beyond that which can be gained by doing research alone. For believing in me and for all the freedom of research I enjoyed, thank you! I am wholehearteadly thankful to Murtuza Jadliwala for being with me every step of the way and for the great work that I had the honour to do with him.

I am sincerely thankful to all my co-authors for the strong support and contributions that are part of this thesis: Kévin Huguenin, István Lám, Florent Lopez, Laurent Bindschaedler, Hossein Manshaei, Maxim Raya, Praveen Kumar and Sudeep Singh. Thank you for your valuable input and feedback. My appreciation goes to the colleagues at Nokia as well, they enriched my research experience in so many ways: Imad Aad, Gianpaolo Perrucci, Valtteri Niemi and Philip Ginzboorg. Thank you for the opportunity to work on relevant problems and the hands-on experience. I am very grateful to my colleagues at PARC, Emiliano De Cristofaro, Julien Freudiger and Ersin Uzun for our very rich interactions and productive environment during my internship.

Many parts of this thesis are owed to open discussions and challenging ideas that have flourished in a great and productive environment. In particular, I am very grateful to my colleagues and friends at LCA: Mathias, Kévin, Nevena, Marcin, Berker, Reza, Jean Louis, Julien, Vincent, Mohamed, Maxim and Julien.

Thank you for challenging and criticising my work. In particular, I am deeply thankful to my officemate Mathias, who shared a desk with me since our first day at EPFL. To our great friendship, to the long hours of intense work and to the even longer hours of great fun, thank you. Furthermore, my sincere gratitude goes to Jacques, Panos, Wojciech, Eric, Giulia, Nathalie, Sergio and Toufic, who shared many special moments and activities with me during these years.

I am deeply grateful to my family who supported and encouraged me throughout this endeavor: Mile, Gordana and Tijana. This thesis would not exist without you. My hearthfelt thanks go to Claudia, who enlightened my path and brought great joy and happiness into my life.

Last but not least, my sincere gratitude goes to the members of my thesis committee, Matthias Grossglauser, Marc Langheinrich, Sebastian Schnorf and Patrick Thiran for their time and effort invested in reviewing this dissertation.

Contents

Contents **xi**

1 Introduction **1**

I Application-Layer Privacy Protection **9**

2 Privacy-Preserving Meeting Scheduling **11**
2.1 Introduction 12
2.2 System Architecture 13
2.3 Privacy Definitions 16
2.4 Privacy-Preserving Scheduling Algorithms 22
2.5 Implementation and Performance Evaluation 32
2.6 User Study 39
2.7 Extensions 44
2.8 Related Work 47
2.9 Summary 48

3 Privacy-Preserving Location Determination **51**
3.1 Introduction 52
3.2 System Architecture 54
3.3 Problem Definition 56
3.4 Privacy-Preserving Location Determination 57
3.5 Implementation and Performance Evaluation 75
3.6 User Study 78
3.7 Extension 80
3.8 Related Work 82
3.9 Summary 84

II Network-Layer Privacy Exposure **85**

4 Privacy of Social Communities in Pervasive Networks **87**
4.1 Introduction . . . 88
4.2 System Architecture . . . 89
4.3 Community Analysis . . . 91
4.4 Results . . . 100
4.5 Related Work . . . 103
4.6 Summary . . . 104

5 Privacy of Social Relationships in Pervasive Networks **105**
5.1 Introduction . . . 106
5.2 System Architecture . . . 107
5.3 Social Interactions and Communities . . . 108
5.4 Relationship Inference Framework . . . 110
5.5 Results . . . 117
5.6 Related Work . . . 123
5.7 Summary . . . 123

III Cross-Layer Approach for Privacy and Information Sharing **125**

6 Adaptive Information-Sharing in Mobile Social Networks **127**
6.1 Introduction . . . 128
6.2 The SPISM Information-Sharing Platform . . . 129
6.3 Study and Data Collection . . . 136
6.4 Analysis and Evaluation . . . 139
6.5 Related Work . . . 145
6.6 Summary . . . 147

7 Conclusion **149**

Bibliography **151**

Index **163**

CV **165**

Chapter 1

Introduction

The way we connect to the Internet has significantly changed in the course of the last decade. Traditional, desktop-based operating systems such as Microsoft Windows represented more than 93% of all Internet-connected devices back in 2000, whereas today, they only represent a small fraction (29%)[66]. Mobile terminals, including smartphones and tablets, represent nowadays 66% of the total number of Internet-connected devices, marking a clear shift from desktop-based to mobile Internet access with multiple co-existing hardware/software vendors and ecosystems.

Mobile devices have evolved from simple "wireless telephones" to powerful computing devices. In addition to basic call and messaging functionality, they enable users to obtain contextualized services and to share information at an unprecedented scale; from location-based queries to services based on shared interests and physical co-presence, mobile devices are now both enablers and gatekeepers of our digital lives.

The amount and diversity of data stored on mobile devices is rapidly increasing, coming from embedded sensors (such as GPSs, gyroscopes, accelerometers, barometers, hygrometers, magnetometers and thermometers [137]), wireless interfaces (cellular, WiFi, Bluetooth, NFC) and from the users themselves (contacts, agenda, media). Mobile applications, developed by both device manufacturers and third-party companies, process large amounts of data in order to present relevant and timely information to the user [73]. For example, mobile social networks may use location data and wireless interfaces in order to infer the type of place and presence of physically co-located neighbors; similarly, banks and cashiers may require access to the NFC sensor in order to authorize a transaction and to verify the identity of the owner.

Although third-party applications need access to certain data, it is extremely important to limit it to only the information that is strictly necessary for each

application to serve its purpose. Moreover, information needs to be protected from not only unauthorized access but from abusive (but authorized) access and proliferation as well. The former being concerned with information security, the latter point represents a more recent and equally important aspect of personal life: information privacy. We refer to "abusive" access as any kind of access to personal information that is authorized but with a different *purpose* or *extent* than the one for which it is intended. This definition is compliant with the European Data Protection Directive[1], where it is stated that (article 6, par. 1, al. (b) and (c)):

> ... personal data must be:
>
> b) collected for specified, explicit and legitimate *purposes*[2] and not further processed...
>
> c) adequate, relevant and *not excessive*[3] in relation to the purposes for which they are collected ...

This is in contrast with an "unauthorized" access, which is an access to any information about the individual for which the entity is not entitled. For example, an application that provides local weather forecast could be authorized to access location data at a city-level granularity, only once every two hours but if such an application accesses continuous location updates – once per minute at street-level granularity – it may constitute a case of abuse of access to personal information, as the accessed data is *excessive* with respect to the purpose. Therefore, it represents a threat to the privacy and it should not be allowed such an unrestricted access to personal information.

The combination of multiple types of information with a diverse set of applications and services makes the mobile device an extremely palatable target for unscrupulous monitoring, interception, tracking and social analysis [75, 81, 107, 120, 38, 105, 22, 16]. In order to protect the mobile users from attacks on their privacy, the research community, as well as private companies including Google, Apple and Microsoft, have been studying approaches and mechanisms that would limit the access to personal information on mobile devices. The reality, however, is that most of these mechanisms are either insufficient or lack appropriate fine-grained functionalities. For instance, the Wall Street Journal [151] found out that, back in 2010, 55% of the 101 scrutinized applications for Android and iOS sent out unique phone IDs without the users' awareness or consent, whereas 47% sent the users' locations as well. A year later, Android was again under scrutiny

[1] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:NOT`

[2] Emphasis added by the author.

[3] Emphasis added by the author.

because of alleged violations of users' privacy due to the usage statistics-collection software developed by Carrier IQ [150]: The application, transparent and impossible to remove for the users, was recording the keystrokes of the users at all times, including when they sent messages and typed passwords. A similar discovery on iOS put Apple under close examination for collecting, without users' knowledge, their whereabouts as a time-stamped record containing the mobile device's location coordinates, as well as all the received signal strengths of the nearby base stations and WiFi access points [7]. These discoveries point out that attacks on mobile users' privacy are real and they concern a significant fraction of the available mobile applications. Therefore, it is crucial to enable users to control and limit, in a more effective way, the information that their devices are leaking. However, the privacy protection mechanisms should not hinder the adoption of novel applications and services, as privacy does not constitute a goal *per se* but rather a necessity. Hence, these mechanisms should be as transparent and unobtrusive as possible for the mobile users.

From a broad perspective, studies and mechanisms concerned with privacy on mobile devices can be described in terms of the following three dimensions:

1. Evaluation of privacy leaks and protection mechanisms: This category encompasses studies that either (i) quantify the extent of the exposure of personal information of mobile users (without taking any action to control it) or (ii) develop mechanisms that protect the users' privacy by actively policing access to information and thus limiting the exposure. An example of the latter is visible in Android, iOS and Windows Phone, where access to private information has to be granted by the user, either at the installation time of the application or when requested for the first time.

2. Centralized and distributed mechanisms: This category includes architectural aspects of the privacy mechanisms that involve either a direct, peer-to-peer communication among mobile devices or a client-server approach where the devices are coordinated by a central entity that oversees all communications among the devices. For instance, most of the current mobile social networks and micro-blogging services such as Facebook, Google+ and Twitter opted for a centralized approach, where the service provider stores all personal information and allows users to specify criteria-based access policies.

3. Network layers at which the mechanisms operate: This category comprises functional aspects of the protection mechanisms that work at either of the ISO/OSI network layers, in particular the link and network (IP) layers,

as they are lowest layers that contain personal[4] or personally identifiable[5] information, such as a name, a biometric sample or an IP or MAC address.

The three above-mentioned dimensions cover a large realm of possible research directions in the field of privacy and mobile computing. The aim of this thesis is, on the one hand, to explore the *breadth* of the privacy challenge in mobile networks in different yet correlated dimensions. On the other hand, we propose novel mechanisms that would go in the *depth* of some specific problems in order to minimize the leakage of personal information for the users.

Our studies span the three dimensions described above, including the evaluation, the definition of novel protection mechanisms and the design of system and network architectures to support them. In addition to the findings and results of this thesis, we also consider the experimental validation of the proposed mechanisms and the real-world evaluation of information leakage as a fundamental asset. That is why all our results are based on either real deployment and user-studies of the proposed mechanisms – implemented as prototypes – or make use of real data from official state agencies and repositories.

Contributions

In this thesis, we address both the practical and theoretical aspects of privacy in mobile networks across the application and network layers of the ISO/OSI stack. First, we study, develop and evaluate novel, efficient privacy-preserving protocols for specific applications present on current mobile devices. Second, we quantify the leakage of hidden behavioral patterns from co-location information that can be inferred from the network layer data, such as the membership of individuals to specific social communities and the types of relationships among them. Third, we design and evaluate a novel cross-layer system that assists users while sharing personal information on mobile devices, by mimicking their own behavior and minimizing undesired information leakage.

Our main contributions are as follows.

1. On the application layer, we identify two significant and recurring challenges in mobile computing: meeting scheduling and optimal meeting location determination. For these two problems, we propose privacy-preserving protocols that provide strong privacy guarantees while retaining a good performance in terms of computation time and memory efficiency – two

[4] Article 2, al. a), of Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:NOT`

[5] Memorandum M-07-16 of the Executive Office of The President, Office of Management and Budget, May 2007, `http://www.whitehouse.gov/sites/default/files/omb/memoranda/fy2007/m07-16.pdf`

aspects that are extremely important for resource-constrained mobile devices. In particular, for the meeting scheduling problem, we design three novel meeting scheduling algorithms that take advantage of the homomorphic properties of well-known cryptosystems, in order to privately and efficiently compute common user availabilities. We also formally outline the privacy requirements in such scheduling applications, and we implement our solutions on real mobile devices. The experimental measurements and analytic results show that the proposed solutions not only satisfy the privacy properties but also fare better, in regard to computation and communication efficiency, compared to other well-known solutions. With respect to the optimal meeting location problem, we propose privacy-preserving algorithms for determining such a location for a group of users. We perform a thorough privacy evaluation of the proposed approaches, under both passive and active adversarial models, by formally quantifying privacy-loss in this setting. Like for the meeting scheduling evaluation, we implement and experimentally evaluate the proposed protocols. We show that they are efficient and can be executed with ease by current mobile devices in just a few seconds. Moreover, the protocols are scalable on the server-side for multiple concurrent executions.

2. On the network layer, we evaluate the exposure of mobile users to social community inference attacks, and we quantify the extent of such exposure by means of a large network deployment on the EPFL campus. During a four-month trial, 80 participants carried mobile devices and were eavesdropped on by an adversarial wireless mesh network on a university campus. We experimentally evaluate the accuracy of reconstructing the communities of mobile users by an adversary that owns a set of wireless sniffing stations. In addition to studying the users' behavior at the community level, we investigate the susceptibility of their pairwise social relationships to inference attacks as well. In particular, we show that an external adversary controlling a wireless mesh network can reconstruct the social communities better than an experimenter who has access to co-location data stored on the mobile devices. Our findings shed light on the potential threat of mobile users to unwarranted or unwanted profiling from mobile cellular network operators.

3. Combining both network and application layers, we design and evaluate a "smart" information-sharing system for mobile social networks. In particular, our machine-learning-based system designed in (i), called *SPISM*, decides in a (semi-)automatic fashion whether to share information (and the level of detail of the information to be shared) with other users or services, based on contextual features and past behavior. The decision-making

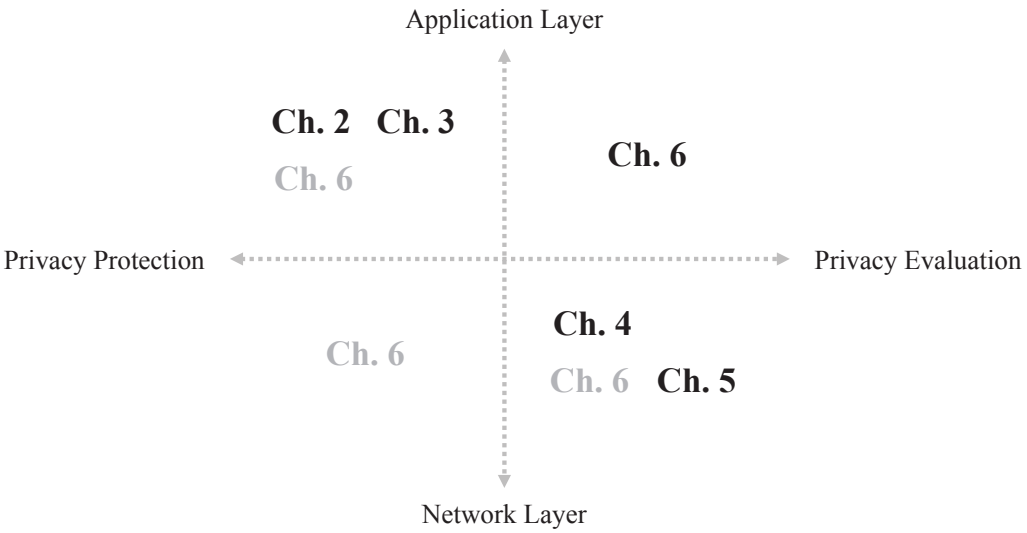


Figure 1.1: Structure of the thesis in terms of the main domains of its chapters. The black-colored chapters are positioned according to their main contribution, whereas the grey-colored chapters are placed according to the domains that they involve.

core is supported by an active learning method that enables SPISM to either decide automatically – whenever the confidence in the decision is high enough – or to rely on the user's input otherwise. SPISM can work with any existing (mobile) online social network and can be used transparently by users, because it can operate at the operating system level, filtering all requests for personal information and replying according to the user's behavior. We show that SPISM significantly outperforms both individual and general user-defined sharing policies, achieving up to 90% of correct sharing decisions, with only a limited cost for the user in terms of initial setup, thanks to active learning. Moreover, we provide insight onto the main reasons behind the sharing decisions, and we show that the type of the requested information, in addition to the social ties of the requester, is an influential feature in the decision process.

Thesis Outline

The structure of the thesis follows the three main contribution areas described above. Figure 1.1 represents the positioning of the different chapters with respect to the dimensions to which they contribute. We introduce the application-layer protection mechanisms in Part I; in particular, we devote Chapter 2 to the meeting-scheduling problem and solutions, whereas in Chapter 3 we define and present our solutions to the optimal meeting-location problem. In Part II we

discuss the network-layer aspects of privacy in mobile networks, starting with the privacy of social communities in Chapter 4 and following with the privacy of social relationships in Chapter 5. In Part III, we present the cross-layer approaches for privacy and, in particular, in Chapter 6 we present our adaptive information-sharing system for mobile social networks.

Publications

Chapter 2 is based on the results presented in [21, 18], whereas Chapter 3 rests on the results contained in [20, 19]. Chapter 4 and Chapter 5 contain the findings presented in [22, 17]. Finally, Chapter 6 is based on the results contained in [16].

Part I

Application-Layer Privacy Protection

Chapter 2

Privacy-Preserving Meeting Scheduling

Mobile devices are used increasingly to store and manage users' personal information, as well as to access popular third-party context-based services. Very often, these applications need to determine common time availabilities among a set of users, in order to enable colleagues, business partners and people to meet. As personal and professional time constraints are often considered as private information, it is crucial to enable users to preserve the privacy of their schedules while they use such applications. In this chapter, we propose practical and privacy-preserving solutions to the server-based scheduling problem for mobile devices. In order to privately and efficiently compute common user availabilities, our three novel algorithms take advantage of the homomorphic properties of well-known cryptosystems. We also formally outline the privacy requirements in such scheduling applications and we implement our solutions on real mobile devices. The experimental measurements and analytical results show that the proposed solutions not only satisfy the privacy properties but also fare better, in regard to computation and communication efficiency, compared to other well-known solutions. Finally, we assess the utility and expectations of the proposed solutions, in terms of privacy and usability, by means of a targeted survey and user-study of mobile-phone users.

Chapter Outline In Section 2.1, we introduce the activity scheduling problem and contrast our approach to existing ones. We then present the system architecture and problem definition in Section 2.2. We formalize the privacy requirements for the scheduling problem in Section 2.3 and outline our algorithms in Section 2.4. We present a comparative analysis and implementation results in Section

2.5, and we summarize the results of our user-study in Section 2.6. We discuss the extensions of our schemes in Section 2.7, and we present the related work in Section 2.8. We summarize this chapter in Section 2.9.

2.1 Introduction

Users rely increasingly on mobile devices such as smartphones, netbooks and lightweight internet tablets to access information while on the move [39], and very often they use the same equipment to store personal information about their daily schedules and activities [34]. Although many context and data sharing applications such as Google Maps, Facebook and Twitter are popular, activity management and synchronization applications are also gaining more and more attention [74]. Applications such as Microsoft Outlook [110], Apple iCal [5] and Nokia Ovi [118] are available on mobile devices, and they all offer time and activity management services. One desirable feature in such applications is activity *scheduling*: Colleagues can schedule meetings at common available time slots; groups of friends can organize parties on weekends, and people unbeknownst to each other can engage in dating based on their common free/busy hours.

One concern in such scheduling applications is that often users prefer not to share all personal information with everyone. For example, they may only want to share common availabilities, but not details about other records. They may also have reservations about sharing personal information with third-party service providers. Therefore, privacy of personal information, *vis-à-vis* service providers and peers, is paramount to the success of such scheduling applications. For instance, a well-known service that allows users to find all common availabilities is Doodle [45]. However, Doodle does not provide privacy: Each user and the doodle server see the free/busy state of all the users, and the private information that is leaked to all users and the central server is well beyond just the common available slots. Cultural, religious and many other private information can be easily inferred from availability patterns. Even if pseudonyms are used instead of real names, the server and all peers still know what time slots are available for everyone and how many users are free or busy.

Privacy-preserving scheduling problems have been extensively studied in the past by researchers from the theoretical perspective, for instance, by modelling them as set-intersection problems [93, 41], distributed constraint-satisfaction problems [155, 158, 143, 142], secure multi-party computation problems [79, 46] and by framing them in the e-voting context [92]. Traditionally, there are two possible approaches to scheduling problems: distributed and centralized. Distributed solutions do not rely on a third-party provider (thus they prevent information from being revealed to the provider), but have several limitations. For instance, due to the frequent and intensive message exchanges among peers, scalability

and computational complexity is an issue when dealing with a large number of (resource-limited) mobile devices; moreover, the need for sequencing among peers and the unpredictability of scheduling results (if a user interrupts the protocol) are two additional drawbacks. The centralized approaches, such as cloud-based computing, are better in terms of scalability, communication cost, complexity, synchronization and resilience but usually do not provide privacy, because users are required to transmit their personal information to the provider.

We provide simple, practical and feasible solutions to the scheduling problem which, in addition to ensuring reasonable privacy guarantees, are easily integrated with existing operational models and mobile service providers. In this chapter, we follow a centralized approach for addressing the problem of efficient and privacy-preserving scheduling. In the proposed schemes, users are able to determine common time slots, without revealing any other information to either the other participants or to the central scheduling server. By building on other works in related domains, we formally define the basic privacy requirements for users in a scheduling scenario. We then propose three novel privacy-preserving scheduling algorithms that take advantage of the homomorphic properties of asymmetric cryptosystems. We implement the proposed algorithms on a test-bed of Nokia mobile devices and perform extensive experiments in order to verify their computation and communication overhead. Moreover, we explain how the system can be further made resilient to collusion and other well-known active attacks. Finally, we present the modalities and results of a targeted user-study on mobile-phone users, focused on both privacy and usability aspects of our applications. To the best of our knowledge, we believe this is the first implementation and extensive testing of privacy-preserving scheduling schemes on commercial mobile devices.

2.2 System Architecture

In this section, we outline the network and adversary model and formally define the scheduling problem.

2.2.1 Network Model

We assume that there is a total of N users u_i, $i \in \{1 \dots N\}$, that want to schedule an activity (meeting, party) at a common available time slot. Each user has a private schedule x_i represented by a string of bits $x_i = [b_{i,1}, b_{i,2}, \dots, b_{i,m}]$, where each bit $b_{i,j} \in \{0, 1\}$ expresses the availability of user u_i in a particular time slot j; $b_{i,j} = 1$ means that user u_i is available at time slot j, whereas $b_{i,j} = 0$ means that the user is not available.[1] We assume that the length m of x_i, i.e. the time

[1] In general, however, users may assign not only a binary value (available or busy) for each time slot, but they could express preferences [50, 59]. For example, $b_{i,j} \in 0, \dots, 10$ where $b_{i,j} = 0$

horizon of the individual schedules, is constant for all users. The value of m can either be pre-decided by the participants or fixed by the application.

Moreover, we assume that each user's device is able to perform public key cryptographic operations and that there is a semi-honest [67] (as detailed in Section 2.2.2) third-party performing the scheduling computations. The latter must be able to communicate with the users and run public key cryptographic functions as well. For instance, a common public-key infrastructure using the RSA [134] cryptosystem could be employed. All communications between a user and the third-party server will be encrypted with the latter's public key for the purposes of confidentiality of the schedules with respect to other users, for authentication and integrity protection. Thus, all users know the public key of the server but nobody, except the server, knows the corresponding private key. For simplicity of exposition, in our algorithms we do not explicitly show the cryptographic operations involving the server's public/private key.

We assume that the N users share a common secret, which is used to derive (i) a fresh common key pair (K_P, K_s), where K_p is the public key and K_s is the private key, and (ii) a fresh bit permutation function $\sigma = [\sigma_1, \ldots, \sigma_m]$ before initiating the scheduling operation. This could be achieved, for example, through a secure credential establishment protocol [29, 32, 101]. Thus, these keys and permutations are derived and known to each member of the group but not to the server. We refer to the encryption of a message M with the group public key as $E_{K_P,r}(M) = C$, where r is a random integer that is eventually needed, and to the decryption of the encrypted message C as $D_{K_s}(C) = M$. The permutation σ, although not strictly required, is used in order to randomize the order of bits sent to the server. This prevents the server from gaining any knowledge about which time slot is being evaluated in each computation.

2.2.2 Adversarial Model

Server The third-party server is assumed to execute the scheduling protocols correctly, but it tries to learn any information it can from the input it gets by the users and the computations it performs. The server can accumulate the knowledge about users in each computation it performs. We refer to this adversarial behavior as *semi-honest*. In most practical settings, where service providers have a commercial interest in providing a faithful service to their customers, the assumption of a semi-honest server is generally sufficient. More details about the semi-honest model can be found in [67].

---

means that user u_i is busy in the time slot j, whereas its preference would increase if $b_{i,j} \geq 1$. For simplicity of exposition, we assume a binary value here. We later discuss a more general case with non-binary costs in Section 2.7.

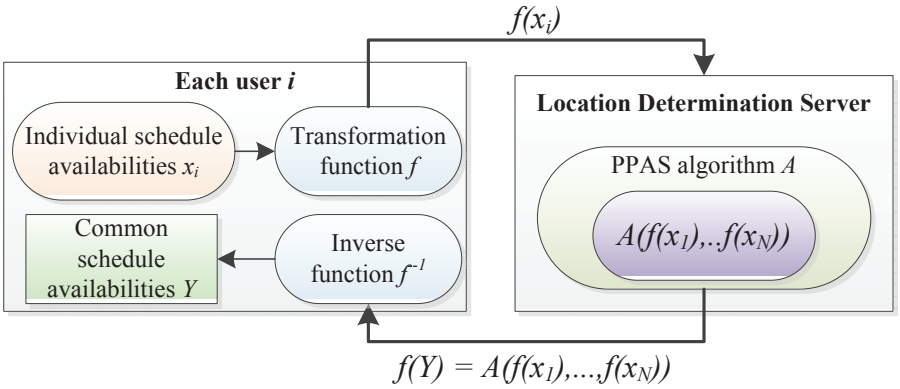


Figure 2.1: Functional diagram of the privacy-preserving activity scheduling (PPAS) protocol, where each user sends his own transformed schedule availabilities to the scheduling server and obtains the aggregate availabilities. The scheduling server obliviously performs the aggregated availabilities, without knowing the individual user schedules.

Users Users also want to learn private information about other users' schedules and, in addition to the passive eavesdropping attacks, users could act maliciously by generating fake users, manipulating their own schedules or by colluding with other users or the scheduling server. Initially, we assume that users are honest but curious (or semi-honest), and afterwards we present more active (or malicious) types of user adversaries in Section 2.7.2.

Although, as mentioned, the semi-honest adversarial model is sufficient in most practical settings, considering the commercial interest of service providers and the mutual trust among participants, it does not include possible malicious behavior by the server or users. For instance, the server could collude with the participants or generate fake participants in order to obtain private information of the participants. Similarly, users might collude with other users or try to maliciously modify their schedules in order to disrupt the execution of the protocol or to gain information about other users' schedules. We address such active attacks by both users and server in Section 2.7.2, and we describe how such attacks can be thwarted by using existing cryptographic mechanisms.

2.2.3 Centralized Scheduling Algorithm

Given a group of N users $u_i, i \in \{1 \dots N\}$, each with private schedules $x_i = [b_{i,1}, \dots, b_{i,m}]$, the scheduling problem is to find time slots j such that $\forall i = 1 \dots N$, $b_{i,j} = 1$, i.e. all users are available in the same time slot j. We refer to an algorithm that solves the scheduling problem as a *scheduling algorithm*. Figure 2.1 shows a functional diagram of a generic privacy-preserving scheduling protocol, where the scheduling algorithm A is executed by a server. Formally, a scheduling

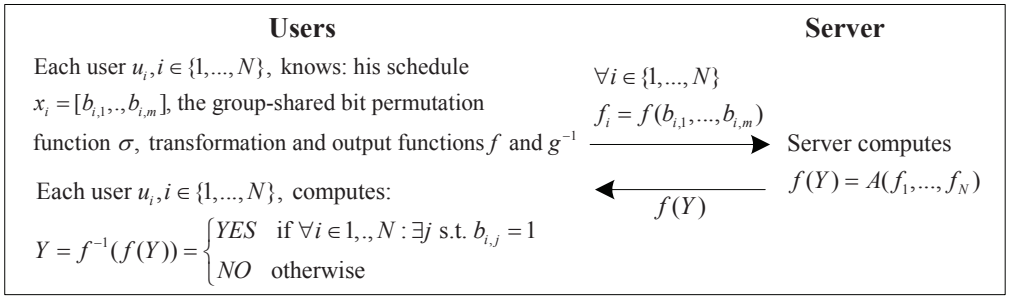


Figure 2.2: A generic scheduling protocol. Users first send their transformed schedules f_i to the server, which then performs the scheduling algorithm A on the received data and sends the encrypted output $f(Y)$ back to each user.

algorithm A accepts the following inputs and produces the respective outputs:

- Input: a transformation of individual schedules

$$f(b_{i,1}, \ldots, b_{i,m}), \quad \forall i = 1 \ldots N.$$

  where f is a one-way public transformation function (based on secret key) such that it is hard (success with only a negligible probability) to determine the input of the function without knowing the secret key, just by observing the output.

- Output: a function $f(Y), Y = y^1, \ldots, y^j, \ldots, y^m$ where:

$$y^j = \begin{cases} YES & \text{if } b_{i,j} = 1, \quad \forall i = 1 \ldots N \\ NO & \text{otherwise} \end{cases}$$

such that each user is able to compute $Y = f^{-1}(f(Y))$ using its local data. As we will see later on, we use the well-known cryptosytems ElGamal [49], Paillier [124] and Goldwasser-Micali [69] as our transformation and output functions f.

A centralized scheduling process works as follows. Each user $u_i, i \in \{1 \ldots N\}$ computes $f_i = f(b_{i,1}, \ldots, b_{i,m})$ and sends it to the third-party server, which then executes the scheduling algorithm A on the received inputs f_i, $\forall i$, and produces $f(Y) = A(f_1, \ldots, f_N)$. Finally, the server sends $f(Y)$ to each user who then obtains $Y = f^{-1}(f(Y))$. Figure 2.2 shows one execution of such a generic centralized scheduling process.

2.3 Privacy Definitions

As mentioned earlier, in this chapter we follow a centralized approach to solve the privacy-preserving scheduling problem. In other words, we assume that a

Table 2.1: Table of symbols.

SYMBOL	DEFINITION
$Adv^{LNK}(A)$	Linkability advantage
$Adv^{IDT}(A)$	Identifiability advantage
$D(C)$	Decryption of a ciphertext C
$E_{K,r}(m)$	Encryption of a message m using the key K and a random number r
K_P	Shared public key of the N users
K_S	Shared private key of the N users
m	Number of slots of each individual schedule
N	Number of users
$x_i=[b_{i,1},..,b_{i,m}]$	Schedule of user u_i, where $b_{i,j}$ is the availability at time slot j
$\sigma=[\sigma_1,..,\sigma_m]$	Schedule permutation function

third-party, given users' individual private schedules, computes their common availabilities (time slots). The privacy provided by a centralized scheduling algorithm can be defined in terms of the following two components: a) User-privacy and b) Server-privacy. Hereafter, we formally define each of these components. The symbols used throughout the chapter are summarized in Table 2.1.

User-privacy

The *user-privacy* of any centralized scheduling algorithm A measures the probabilistic advantage that any user $u_i, i \in \{1 \dots N\}$ gains towards learning the private schedules of at least one other user $u_j, j \neq i$, except their common availabilities, after all users have participated in the execution of the algorithm A. In order to accurately measure users' privacy, we need to compute the following two advantages. First, we measure the *Identifiability Advantage*, which is the probabilistic advantage of an adversary in correctly guessing a schedule bit (which is not a common availability) of any other user. We denote it as $Adv_{u_i}^{IDT}(A)$. Second, we measure the *Linkability Advantage*, which is the probabilistic advantage of an adversary in correctly guessing that any two or more other users have exactly the same corresponding schedule bit (not a common availability bit) without necessarily knowing the values of those bits. We denote this advantage as $Adv_{u_i}^{LNK}(A)$. We make the following straightforward observation.

Observation 1. *If an adversary has identifiability advantage over two corresponding schedule bits of two different users, this implies that it has linkability advantage over those two bits as well. However, the inverse is not necessarily true.*

We semantically define the identifiability and linkability advantages using a challenge-response methodology. Challenge-response games have been widely

used in cryptography to prove the security of cryptographic protocols. We now describe such a challenge-response game for the identifiability advantage $Adv_{u_i}^{IDT}(A)$ of any user u_i participating in the algorithm A as follows.

1. Initialization: Challenger privately collects $x_i = [b_{i,1}, \ldots, b_{i,m}]$ and $f_i = f(b_{i,1}, \ldots, b_{i,m})$ from all users $u_i, i \in \{1 \ldots N\}$.

2. Scheduling: Challenger computes $f(Y) = A(f_1, f_2, \ldots, f_N)$ with the users and sends $f(Y)$ to all users $u_1, u_2, \ldots, u_N$.

3. Challenger randomly picks a user u_i, $i \in \{1 \ldots N\}$, as the adversary.

4. u_i picks $j \in \{1 \ldots N\}$, $s.t.\ j \neq i$ and sends it to the challenger.

5. Challenge: the challenger picks a random time slot $p \in \{1 \ldots m\}$, $s.t.$, $\exists b_{k,p} = 0$ for at least one $k \in 1, \ldots, N$. Challenger then sends (j, p) to the user u_i. This is the challenge.

6. Guess: User u_i sends $b'_{j,p} \in \{0, 1\}$ to the challenger as a response to his challenge. If $b'_{j,p} = b_{j,p}$, the user u_i (adversary) wins; otherwise, he loses.

The identifiability advantage $Adv_{u_i}^{IDT}(A)$ can be defined as

$$Adv_{u_i}^{IDT}(A) = \left| Pr_{u_i}[b'_{j,p} = b_{j,p}] - \frac{1}{2} \right| \tag{2.1}$$

where $Pr_{u_i}[b'_{j,p} = b_{j,p}]$ is the probability of user u_i winning the game (correctly answering the challenge in the challenge-response game), computed over the coin flips of the challenger, $b'_{j,p}$ is u_i's guess about the schedule of user u_j in the time slot p and $b_{j,p}$ is u_j's true availability. We note that Eq. 2.1 defines the probabilistic advantage of u_a in a single execution of the challenger-adversary game, and therefore it does not capture the additional knowledge that could be gained by repeated executions of the same game with different subsets of the participants. This assumption is formalized in Definition 2.4. An external attacker, having no access to the output of the algorithm, has obviously no advantage at all. Thus, we focus on the non-trivial case with participating users only.

Similarly, we describe the challenge-response game for the linkability advantage $Adv_{u_i}^{LNK}(A)$ of any user u_i as follows.

1. Initialization: Challenger privately collects $x_i = [b_{i,1}, \ldots, b_{i,m}]$ and $f_i = f(b_{i,1}, \ldots, b_{i,m})$ from all users $u_i, i \in \{1 \ldots N\}$.

2. Scheduling: Challenger computes $f(Y) = A(f_1, f_2, \ldots, f_N)$ with the users and sends $f(Y)$ to all users $u_1, u_2, \ldots, u_N$.

3. Challenger randomly picks a user u_i, $i \in \{1 \ldots N\}$, as the adversary.

4. u_i picks $h, j \in \{1 \dots N\},\ s.t.\ j \neq h, j \neq i, h \neq i$ and sends (h, j) to the challenger.

5. Challenge: Challenger randomly picks a time slot $p \in \{1 \dots m\},\ s.t.,\ \exists b_{k,p} = 0$ for at least one $k \in 1, \dots, N$. Challenger then sends (j, p) and (h, p) to the user u_i. This is the challenge.

6. Guess: User u_i decides if $b_{j,p} = b_{h,p}$ or not. User u_i sets $b' = 1$ if he decides $b_{j,p} = b_{h,p}$ and $b' = 0$ if he decides $b_{j,p} \neq b_{h,p}$. User u_i sends b' to the challenger as a response to his challenge. If $b_{j,p} = b_{h,p}$ and $b' = 1$ or if $b_{j,p} \neq b_{h,p}$ and $b' = 0$, the user u_i (adversary) wins; otherwise, he loses.

The linkability advantage $Adv_{u_i}^{LNK}(A)$ can be defined as

$$Adv_{u_i}^{LNK}(A) = \left| Pr_{u_i}[((b_{j,p} = b_{h,p}) \ \wedge \ b' = 1) \vee \ ((b_{j,p} \neq b_{h,p}) \ \wedge \ b' = 0)] - \frac{1}{2} \right|$$

where $Pr_{u_i}[.]$ is the probability of user u_i winning the game, computed over the coin flips of the challenger. As for the identifiability advantage, an external attacker has no linkability advantage at all.

We now define the user-privacy of the scheduling algorithm A on a per-execution basis as follows:

Definition 2.1. *An execution of the centralized scheduling algorithm A is* user-private *if both the identifiability advantage $Adv_{u_i}^{IDT}(A)$ and the linkability advantage $Adv_{u_i}^{LNK}(A)$ of each participating user $u_i, i \in \{1, \dots, N\}$ is negligible.*

A function $f(x)$ is called *negligible* if, for any positive polynomial $p(x)$, there is an integer B such that for any integer $x > B$, $f(x) < 1/p(x)$ [67].

Definition 2.1 says that a particular execution of the scheduling algorithm is user-private if and only if users do not gain any (actually, negligible) additional knowledge about the schedule bits of any other user, except the schedule bits that have a value 1 for all users (common availabilities).

Server-privacy

The *server-privacy* of any (centralized) scheduling algorithm A measures the probabilistic advantage that the server (which executes the scheduling algorithm A and observes the inputs from the users) gains towards learning the private schedules of at least one user $u_i, i \in \{1 \dots N\}$. As in the case of user-privacy, we need to compute the following two advantages. First, the advantage of the server in guessing correctly any schedule bit of any user participating in the scheduling algorithm, called as *Identifiability Advantage* and denoted as $Adv_S^{IDT}(A)$. Second, the advantage of the server in guessing correctly that any two (or more) participating users have exactly the same corresponding schedule bits without

necessarily knowing the values of those bits, called the *Linkability Advantage* and denoted as $Adv_S^{LNK}(A)$.

The server identifiability and linkability advantages are defined in a similar fashion as the user advantages. The challenge-response game for the server identifiability advantage $Adv_S^{IDT}(A)$ is defined as follows.

1. Initialization: Challenger privately collects $x_i = [b_{i,1}, \ldots, b_{i,m}]$ and the server privately collects $f_i = f(b_{i,1}, \ldots, b_{i,m})$ from all users $u_i, i \in \{1 \ldots N\}$.

2. Scheduling: Server computes $f(Y) = A(f_1, f_2, \ldots, f_N)$ with the users and sends $f(Y)$ to all users $u_1, u_2, \ldots, u_N$.

3. Server picks $i \in \{1 \ldots N\}$ and sends it to the challenger.

4. Challenge: Challenger randomly picks a time slot $p \in \{1 \ldots m\}$. Challenger then sends (i, p) to the server. This is the challenge.

5. Guess: server sends $b'_{i,p} \in \{0, 1\}$ to the challenger as a response to his challenge. If $b'_{i,p} = b_{i,p}$, the server (adversary) wins; otherwise, he loses.

The identifiability advantage $Adv_S^{IDT}(A)$ is defined as

$$Adv_S^{IDT}(A) = \left| Pr_S[b'_{j,p} = b_{j,p}] - \frac{1}{2} \right| \tag{2.2}$$

where $Pr_S[b'_{j,p} = b_{j,p}]$ is the probability of the server winning the game, computed over the coin flips of the challenger.

The challenge-response game for the server linkability advantage $Adv_S^{LNK}(A)$ is defined as follows.

1. Initialization: Challenger privately collects $x_i = [b_{i,1}, \ldots, b_{i,m}]$ and the server privately collects $f_i = f(b_{i,1}, \ldots, b_{i,m})$ from all users $u_i, i \in \{1 \ldots N\}$.

2. Scheduling: Server computes $f(Y) = A(f_1, f_2, \ldots, f_N)$ with the users and sends $f(Y)$ to all users $u_1, u_2, \ldots, u_N$.

3. Server picks $h, j \in \{1 \ldots N\},\ s.t.\ j \neq h$ and sends (h, j) to the challenger.

4. Challenge: Challenger randomly picks $p \in \{1 \ldots m\}$ and then sends (j, p) and (h, p) to the server. This is the challenge.

5. Guess: Server decides if $b_{j,p} = b_{h,p}$ or not. Server sets $b' = 1$ if he decides $b_{j,p} = b_{h,p}$ and $b' = 0$ if he decides $b_{j,p} \neq b_{h,p}$. Server sends b' to the challenger as a response to his challenge. If $b_{j,p} = b_{h,p}$ and $b' = 1$ or if $b_{j,p} \neq b_{h,p}$ and $b' = 0$, the server (adversary) wins; otherwise, he loses.

The linkability advantage $Adv_S^{LNK}(A)$ is defined as

$$Adv_S^{LNK}(A) = \left| Pr_S[(b_{j,p} = b_{h,p}) \ \wedge \ b' = 1) \vee \ (b_{j,p} \neq b_{h,p}) \ \wedge \ b' = 0)] - \frac{1}{2} \right|$$

where $Pr_S[.]$ is the probability of the server winning the game, computed over the coin flips of the challenger.

The server-privacy of the scheduling algorithm A on a per-execution basis can then be defined as follows:

Definition 2.2. *An execution of the centralized scheduling algorithm A is* server-private *if both the identifiability advantage $Adv_S^{IDT}(A)$ and the linkability advantage $Adv_S^{LNK}(A)$ of the server is negligible.*

Now, it is reasonable to assume that in practice users will be able to perform multiple executions of the scheduling algorithm with possibly different participating sets of users. This is especially true if such an algorithm is offered, for example, as a service by mobile service providers to their subscribers. Thus, privacy of the scheduling algorithm should be defined over multiple executions. First, we define a *private execution* as follows:

Definition 2.3. *A* private *execution is an execution which does not reveal more information than what can be derived from its result and the prior knowledge.*

Based on how memory is retained over sequential executions, we define two types of algorithm executions, namely, independent and dependent:

Definition 2.4. *An* independent (respectively, dependent) execution *is a single private execution of the scheduling algorithm defined in Section 2.2.3 in which* no (respectively, some) *information of an earlier and current execution is retained and passed to a future execution.*

The information retained can include past inputs to the algorithm, intermediate results (on the server) and the outputs of the algorithm. Based on the type of executions, we define a privacy-preserving scheduling algorithm as follows:

Definition 2.5. *A scheduling algorithm A is* execution *(respectively* fully*)* privacy-preserving *if and only if for every* independent *(respectively* all*) execution(s):*

1. *A is correct; All users are correctly able to compute $y^j = 1, \forall j = 1 \ldots m$ if and only if $b_{i,j} = 1, \forall i = 1 \ldots N$.*

2. *A is user-private in every execution.*

3. *A is server-private in every execution.*

A fully privacy-preserving algorithm is a much stronger (and difficult to achieve) privacy requirement. In this work, similar to earlier efforts, we focus on achieving execution privacy. The following observation gives the relationship between fully privacy-preserving and execution privacy-preserving scheduling algorithms.

Observation 2. *Any scheduling algorithm A, as defined in Section 2.2.3, is execution privacy-preserving if it is fully privacy-preserving. However, the inverse is not true.*

Next, we outline our centralized scheduling algorithms.

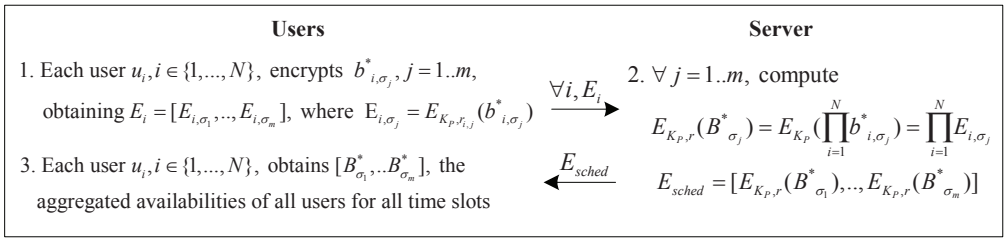


Figure 2.3: *SchedElg* protocol.

2.4 Privacy-Preserving Scheduling Algorithms

In this section, we present our three privacy-preserving scheduling algorithms. For each algorithm, we first outline the basic cryptographic properties that are used, and then we describe and show their operational mechanisms in detail. We finally state the privacy guarantees provided by each of the algorithms.

2.4.1 SchedElG

Our first privacy-preserving centralized scheduling scheme is based on the ElGamal [49] cryptosystem. The security of the ElGamal encryption relies on the intractability of the discrete logarithm problem (DLP), which assumes that it is computationally infeasible to obtain the private key K_s given the public key (g, h), where g is a generator of a multiplicative cyclic group G of prime order q and $h = g^{K_s} \mod q$.

Our protocol *SchedElG* uses the *homomorphic* property of the ElGamal cryptosystem in order to allow the scheduling server to compute the aggregated availabilities by working only on the encrypted individual schedules. For instance, it can be verified that the ElGamal scheme satisfies:

$$D(E_{K_P,r_1}(m_1) \cdot E_{K_P,r_2}(m_2)) = D((g^{r_1}, m_1 h^{r_1}) \cdot (g^{r_2}, m_2 h^{r_2})) = D(g^r, (m_1 \cdot m_2) h^r)$$
$$= m_1 \cdot m_2$$

where $r = r_1 + r_2 \in \mathbb{Z}_q$ is a random integer. Moreover, being a probabilistic encryption scheme, it follows that if $r_1 \neq r_2$, $E_{K_P,r_1}(m) \neq E_{K_P,r_2}(m)$.

For the *SchedElG* algorithm, we assume that the meeting participants represent their availabilities in the following way: $b^*_{i,j} = 1$ if $b_{i,j} = 1$, but $b^*_{i,j} = R$ (where $R \in \mathbb{Z}_q, R > 1$ is a random integer) if $b_{i,j} = 0$.

Scheme

The privacy-preserving scheduling protocol *SchedElG* is shown in Figure 2.3. All users first select the sequence of time slots according to the permutation σ, i.e., σ_j, $\forall j = 1..m$, and then encrypt individually the corresponding schedule availabilities, i.e., $E_i = [E_{i,\sigma_1}, \ldots, E_{i,\sigma_m}]$ where $E_{i,\sigma_j} = E_{K_P,r_{i,j}}(b^*_{i,\sigma_j})$. Then, each user sends its E_i privately to the scheduling server that performs the multiplication $\prod_{i=1}^{N} E_{i,\sigma_j}$ of all users' encrypted schedules E_{i,σ_j}, for $j = 1, \ldots, m$. The results of such operation are the (encrypted) aggregated availabilities of all users for each time slot j. Next, the server replies with the aggregated encrypted result E_{sched} back to each user. Each slot in E_{sched} contains a product of the individual time-slot bits encrypted with the users' common session key. Finally, each user decrypts the result and obtains the aggregated availabilities $[y^1 = B^*_{\sigma_1}, \ldots, y^m = B^*_{\sigma_m}]$ of all users u_i for each time slot σ_j. If $B^*_{\sigma_j} = 1$, it means that all users are available at time slot σ_j; if $B^*_{\sigma_j} > 1$, then at least one user is not available and therefore σ_j is not a suitable time slot. The following result shows the correctness and privacy properties of *SchedElG*.

Lemma 2.1. *The protocol* SchedElG *is correct and execution privacy-preserving.*

Proof. **Correctness** From Section 2.2.3, we know that any scheduling algorithm should output $f(Y)$, on inputs $f_1, f_2, \ldots, f_N$, where $f_i = f(b_{1,1}, \ldots, b_{i,m})$, such that each user is able to privately compute $Y = f^{-1}(f(Y))$, where $Y = y^1, \ldots, y^j, \ldots, y^m$. The output bit $y^j, \forall j$ should be such that it should take some value v if and only if all users are available. Otherwise, the output bit y^j never takes value v and should take some other value, indicating that at least one user is not available. From Figure 2.3, we can see that, provided the homomorphic properties of the ElGamal cryptosystem are correct, we have that (with overwhelming probability) $y^j = 1$ if and only if $b_{i,j} = 1, \forall i$, i.e., all users are available. Otherwise we have $y^j = R$, where $R > 1$ is some random number. Thus, *SchedElg* is correct.

Privacy

In order to be *user-private*, the identifiability and linkability advantages defined in Section 2.2 must be a negligible function. Formally, we need that

$$Adv_{u_i}^{IDT}(SchedElG) = \left| Pr_{u_i}[b'_{j,p} = b_{j,p}] - \frac{1}{2} \right| < \frac{1}{p(N)}$$

$$Adv_{u_i}^{LNK}(SchedElG) = \left| Pr_{u_i}[(b_{j,p} = b_{h,p}) \ \wedge \ b' = 1) \ \vee \ (b_{j,p} \neq b_{h,p}) \ \wedge \ b' = 0)] - \frac{1}{2} \right| < \frac{1}{p(N)}$$

where $Pr_{u_i}[b'_{j,p} = b_{j,p}]$ and $Pr_{u_i}[(b_{j,p} = b_{h,p}) \wedge b' = 1) \vee (b_{j,p} \neq b_{h,p}) \wedge b' = 0)]$ are the probabilities of a user u_i winning the challenge-response games, and $p(N)$ is any positive polynomial function of N. Without loss of generality, we assume that the Challenger chooses user u_1 as the Adversary. Moreover, as the computation of the availabilities for all time slots are identical, we provide the proof for one time slot p only.

Hereafter we provide the privacy proofs for both client- and server-privacy, by computing the respective identifiability and linkability advantages.

- User identifiability advantage
  After Step 4 of the challenger-response game, u_1 knows (i) its own schedule bit $b_{1,p}$ and (ii) the non-trivial result of the algorithm $B_p^* = b_{1,p}^* \cdot \ldots \cdot b_{N,p}^* > 1$, i.e. there is at least one user that is not available in the time slot p. Therefore, the identifiability advantage becomes

  $$Adv_{u_i}^{IDT}(SchedElG) = \left| Pr_{u_i}[b'_{j,p} = b_{j,p} | B_p^* > 1, b_{1,p}] - \frac{1}{2} \right|$$

  where

  $$\begin{aligned}
  & Pr_{u_i}[b'_{j,p} = b_{j,p} | B_p^* > 1, b_{1,p}] \\
  &= \sum_{k=0}^{1} Pr(b'_{j,p} = b_{j,p} | B_p^* > 1, b_{1,p} = k) \cdot Pr(b_{1,p} = k | B_p^* > 1) \\
  &= \sum_{k=0}^{1} \sum_{z=0}^{1} Pr(b'_{j,p} = z \wedge b_{j,p} = z | B_p^* > 1, b_{1,p} = k) \cdot Pr(b_{1,p} = k | B_p^* > 1) \\
  &= \sum_{k=0}^{1} \sum_{z=0}^{1} Pr(b'_{j,p} = z | B_p^* > 1, b_{1,p} = k) \cdot Pr(b_{j,p} = z | B_p^* > 1, b_{1,p} = k) \cdot Pr(b_{1,p} = k | B_p^* > 1)
  \end{aligned}$$

  Given that the Challenger chooses a time slot p where $\exists b_{q,p} = 0, q \in$

$\{1, \ldots, N\}$, we have

$$Pr(b'_{j,p} = 0 | B^*_p > 1, b_{1,p} = 0) = Pr(b'_{j,p} = 1 | B^*_p > 1, b_{1,p} = 0) = 1/2$$
$$Pr(b_{j,p} = 0 | B^*_p > 1, b_{1,p} = 0) = Pr(b_{j,p} = 1 | B^*_p > 1, b_{1,p} = 0) = 1/2$$

$$\begin{aligned} Pr(b'_{j,p} = 0 | B^*_p > 1, b_{1,p} = 1) &= Pr(b_{j,p} = 0 | B^*_p > 1, b_{1,p} = 1) \\ &= \frac{\sum_{m=1}^{N-1} C_m^{N-1} \cdot m}{(2^{N-1} - 1) \cdot (N-1)} = \frac{2^{N-2}}{2^{N-1} - 1} \\ Pr(b'_{j,p} = 1 | B^*_p > 1, b_{1,p} = 1) &= Pr(b_{j,p} = 1 | B^*_p > 1, b_{1,p} = 1) \\ &= \frac{\sum_{m=1}^{N-2} C_m^{N-1} \cdot m}{(2^{N-1} - 1) \cdot (N-1)} = \frac{2^{N-2} - 1}{2^{N-1} - 1} \end{aligned}$$

which implies

$$Pr_{u_i}[b'_{j,p} = b_{j,p} | B^*_p > 1, b_{1,j}] = \frac{a}{2} + (1-a) \cdot \frac{2^{2(N-2)} + (2^{N-2} - 1)^2}{(2^{N-1} - 1)^2}$$

where $a = Pr(b_{1,p} = 0 | B^*_p > 1)$. By including this result, we have that

$$Adv^{IDT}_{u_i}(SchedElG, N) = \left| \frac{a}{2} + (1-a) \cdot \underbrace{\frac{2^{2(N-2)} + (2^{N-2} - 1)^2}{(2^{N-1} - 1)^2}}_{\gamma} - \frac{1}{2} \right|$$

where

$$\begin{aligned} \gamma &= \frac{2^{2N-4} + 2^{2N-4} - 2 \cdot 2^{N-2} + 1}{2^{N-2} - 2 \cdot 2^{N-1} + 1} = \frac{2^{2N-3} - 2^{N-1} + 1}{2^{2N-2} - 2^N + 1} \\ &= \frac{(2^{2N-2} - 2^N + 1) + 1}{2 \cdot (2^{2N-2} - 2^N + 1)} = \frac{1}{2} + \frac{1}{2(2^{2N-2} - 2^N + 1)} \end{aligned}$$

By combining the previous expressions, we obtain

$$\begin{aligned} Adv^{IDT}_{u_i}(SchedElG, N) &= \left| \frac{a}{2} + (1-a) \cdot \gamma - \frac{1}{2} \right| = \left| \frac{1-a}{2^{2N-1} - 2^{N+1} + 2} \right| \\ &= \left| \frac{1-a}{2^{N+1}(2^{N-2} - 1) + 2} \right| \overset{\forall N > 2}{<} \frac{1}{2^N} \end{aligned}$$

which holds $\forall 0 \leq a \leq 1$. Therefore $Adv^{IDT}_{u_i}(SchedElG, N)$ is a negligible function of the number of participants N, as it approaches zero faster than the reciprocal of any polynomial, for large enough N [11].

- User linkability advantage
  By definition we have

$$Adv^{LNK}_{u_i}(SchedElG)$$
$$= \left| Pr_{u_i}[(b_{j,p} = b_{h,p}) \wedge\ b' = 1)\ \vee\ (b_{j,p} \neq b_{h,p})\ \wedge\ b' = 0) | B^*_p > 1, b_{1,p}] - \frac{1}{2} \right|$$

From the above, we obtain

$$Pr_{u_i}[(b_{j,p} = b_{h,p}) \wedge\ b' = 1)\ \vee\ (b_{j,p} \neq b_{h,p})\ \wedge\ b' = 0) | B_p^* > 1, b_{1,p}] =$$
$$\sum_{k=0}^{1} Pr[(b_{j,p} = b_{h,p}) \wedge\ b' = 1 | B_p^* > 1, b_{1,p} = k)] \cdot Pr(b_{1,p} = k | B_p^* > 1) +$$
$$\sum_{k=0}^{1} Pr[(b_{j,p} \neq b_{h,p}) \wedge\ b' = 0 | B_p^* > 1, b_{1,p} = k)] \cdot Pr(b_{1,p} = k | B_p^* > 1)$$

which implies

$$Pr_{u_i}[(b_{j,p} = b_{h,p}) \wedge\ b' = 1)\ \vee\ (b_{j,p} \neq b_{h,p})\ \wedge\ b' = 0) | B_p^* > 1, b_{1,p}] =$$
$$\frac{a}{2} + (1-a) \cdot \left\{ \left[\left(\frac{2^{N-2}}{2^{N-1}-1} \right)^2 + \frac{2^{N-3}-1}{2^{N-2}} \cdot \frac{2^{N-2}-1}{2^{N-1}-1} \right]^2 + \left[\frac{1}{4} + \frac{1}{2} \frac{2^{N-2}-1}{2^{N-1}-1} \right]^2 \right\}$$

where $a = Pr(b_{1,p} = 0 | B_p^* > 1)$. Similarly to the identifiability advantage, it can be shown that $Adv_{u_i}^{LNK}(SchedElG, N)$ is a negligible function of the number of participants N. As both identifiability and linkability advantages are negligible functions (in the number of participants N), *SchedElG* is user-private.

- Server advantages
  The server that is performing the computations on the encrypted schedules does not know any user's schedule bit, as all schedules have been encrypted by the users prior to being sent to the server with the users' shared public key, and only they know the corresponding private key. Therefore, $Adv_S^{IDT}(SchedElG) = Adv_S^{LNK}(SchedElG) = 0$, i.e. *SchedElG* is server-private. ■

□

For illustration purposes, in Figure 2.4 we plotted the identifiability and linkability advantages of an adversary for *SchedElg*, compared with polynomially (in terms of the number of participants N) decreasing functions $1/p(N)$. As confirmed by our analysis, the plot shows that both identifiability and linkability advantages are lower than the considered polynomials $1/p(N)$, for the given values of N.

2.4.2 SchedPa Algorithm

In this section, we define our second privacy-preserving scheduling scheme, which is based on the Paillier cryptosystem [124]. The security of the Paillier encryption scheme is based on the intractability of determining whether an integer r is an n-residue mod n^2, where n is a composite number. In our protocol, we use the

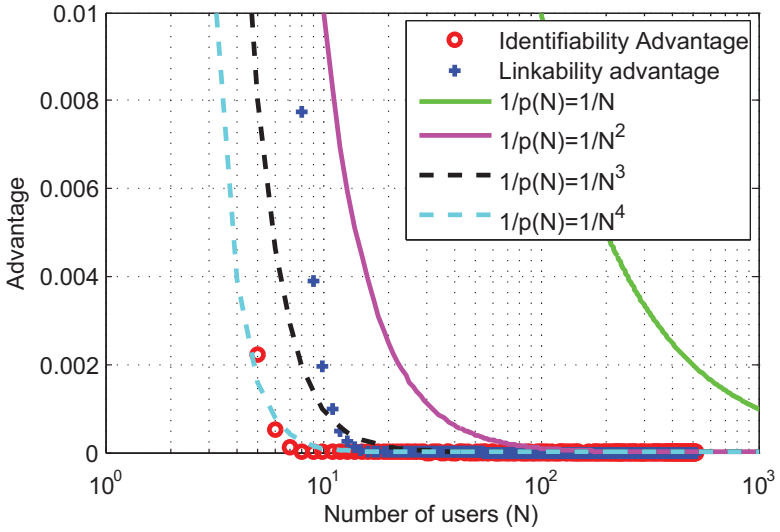


Figure 2.4: Identifiability and linkability advantages of an adversary.

homomorphic properties of the Paillier cryptosystem to compute in a privacy-preserving fashion the availability of all users involved in the scheduling process. In particular, one can verify that the Paillier scheme satisfies the following:

$$D[E_{K_P,r_1}(m_1) \cdot E_{K_P,r_1}(m_2) \mod n^2] = m_1 + m_2 \mod n$$
$$D[E_{K_P,r}(m_1)^{m_2} \mod n^2] = m_1 \cdot m_2 \mod n$$

where $r_i, r \in \mathbb{Z}_n^*$ are random numbers chosen by the encrypters, $m \in \mathbb{Z}_n$ is the message to encrypt and $n = pq$ where p, q are two large primes. The randomness in the encryption ensures that if $r_1 \neq r_2$, $E_{K_P,r_1}(m) \neq E_{K_P,r_2}(m)$.

To adapt our scheme to the addition property of Paillier's homomorphism, we take the bit value $\bar{b}_{i,j}$ in the computation instead of the original bit value $b_{i,j}$ as follows: $\bar{b}_{i,j} = 0$ if $b_{i,j} = 1$, and $\bar{b}_{i,j} = r$ (where $r \in \mathbb{Z}_n^*, r > 1$ is a random integer) if $b_{i,j} = 0$.

Scheme

The corresponding privacy-preserving scheduling protocol is shown in Figure 2.5. First, all users select the sequence of time slots according to the permutation σ, i.e., σ_j, $\forall j = 1, \ldots, m$, and then encrypt individually the corresponding availabilities, i.e. $E_i = [E_{i,\sigma_1}, \ldots, E_{i,\sigma_m}]$ where $E_{i,\sigma_j} = E_{K_P,r_{i,j}}(\bar{b}_{i,\sigma_j})$. Then, each user sends its E_i privately to the scheduling server that performs the multiplication and exponentiation $(\prod_{i=1}^{N} E_{i,\sigma_j})^R$ of all users' encrypted schedules E_{i,σ_j}, for $j = 1, \ldots, m$, in order to obtain the encryption of the value V_{σ_j} that is needed by the users. Afterwards, the server sends the aggregated encrypted result E_{sched} back to each user. Each slot in E_{sched} contains a randomly scaled sum of the

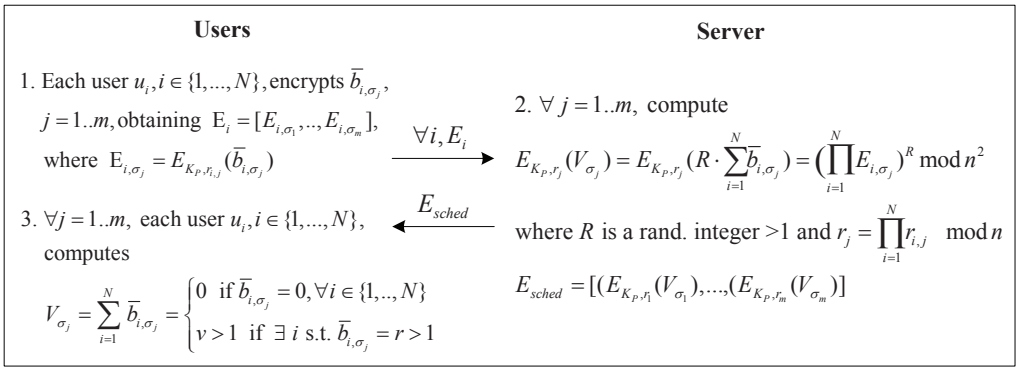


Figure 2.5: *SchedPa* protocol.

individual time-slot bits $\overline{b}_{i,\sigma_j}$ encrypted with the users' common session key. Finally, each user decrypts the result and knows that if $V_{\sigma_j} = 0$, the time slot σ_j is available for everybody. If $V_{\sigma_j} > 1$, then at least one user is not available. Note that even if the server chooses $R = 1$, the privacy of the users is preserved with $\overline{b}_{i,j}$. The following result shows the correctness and privacy properties of *SchedPa*.

Lemma 2.2. *The protocol* SchedPa *is correct and execution privacy-preserving.*

Proof. **Correctness** From Section 2.2.3, we know that any scheduling algorithm should output $f(Y)$, on inputs $f_1, f_2, \ldots, f_N$, where $f_i = f(b_{1,1}, \ldots, b_{i,m})$, such that each user is able to privately compute $Y = f^{-1}(f(Y))$, where $Y = y^1, \ldots, y^j, \ldots, y^m$. The output bit $y^j, \forall j$ should be such that it should take some value v if and only if all users are available. Otherwise, the output bit y^j never takes value v and should take some other value, indicating that at least one user is not available. From Figure 2.3, we can see that, provided the homomorphic properties of the Paillier cryptosystem are correct, we have that (with overwhelming probability) $y^j = 0$ if and only if $b_{i,j} = 1, \forall i$, i.e., all users are available. The value of $y^j = R$, where $R > 1$ is some random number, otherwise. Thus, *SchedPa* is correct.

Privacy

Hereafter we present the privacy proofs, both for user- and server-privacy.

- User advantages
  The knowledge that any user u_i has in the *SchedPa* game is the same as in *SchedElG*. In particular, u_i knows that $V_p = R \cdot \sum_{k=1}^{N} b_{k,p} > 0$ and therefore it knows that there is at least one user u_k, $k \in \{1, \ldots, N\}$ that is not available in the time slot p. Moreover, each user u_i knows its own schedule $b_{i,p}$. As a consequence, $Adv_{u_i}^{IDT}(SchedPa) = Adv_{u_i}^{IDT}(SchedElG)$

and $Adv_{u_i}^{LNK}(SchedPa) = Adv_{u_i}^{LNK}(SchedElG)$ and therefore *SchedPa* is user-private.

- Server advantages
  As in the *SchedElG* algorithm, the server performing the *SchedPa* algorithm does not have access to any schedule bit and therefore *SchedPa* is server-private. ■

□

2.4.3 SchedGM Algorithm

In this section, we present our third privacy-preserving scheduling algorithm, which is based on the Goldwasser-Micali (GM) cryptographic scheme [69]. The security of the GM encryption relies on the intractability of the quadratic residuosity problem, i.e. on the infeasibility of determining whether or not an integer r is a quadratic residue mod n when the Jacobi symbol for r is 1, given $n = pq$ where p, q are large primes. *SchedGM* makes use of the following homomorphic property of the GM cryptosystem:

$$D[E_{K_P,r_1}(m_1) \cdot E_{K_P,r_2}(m_2)] = m_1 \oplus m_2$$

The intuition behind the protocol is based on the work by Herlea *et al.* [79], in which users privately establish a global bit mask (unknown to any user) and then compare all the masked availabilities without knowing the true bit value b_{i,σ_j} of the other users. If all users have the same masked bit value for a given time slot σ_j, then each user knows that everybody else has the same availability, which can be inferred by looking at the private unmasked bit value b_{i,σ_j}. Although initially used in a distributed scenario, we extend the general idea to the centralized scheme as well.

Assumption

Each user u_i generates a private random bit mask $s_i = [c_{i,1}, c_{i,2}, \ldots, c_{i,m}], c_{i,j} \in \{0, 1\}$, of the same length of the schedule x_i.

Scheme

The privacy-preserving scheduling algorithm is shown in Figure 2.6. Each user first selects the sequence of time slots according to the permutation σ, i.e., σ_j, $\forall j = 1, \ldots, m$, and then masks the corresponding schedule bits, i.e. $b_{i,\sigma_j}^{\oplus} = b_{i,\sigma_j} \oplus c_{i,j}$. Then, each user encrypts individually both its bit mask, i.e. $E_i^c = [E_{K_P,r_{i,1}}(c_{i,1}), \ldots, E_{K_P,r_{i,m}}(c_{i,m})$, and the masked availabilities, i.e. $E_i = [E_{i,\sigma_1}, \ldots, E_{i,\sigma_m}]$, where $E_{i,\sigma_j} = E_{K_P,r_{i,j}}(b_{i,\sigma_j}^{\oplus})$. Afterwards, each user u_i sends its E_i and E_i^c to the server, which computes the multiplication of the received E_{i,σ_j} with the encrypted

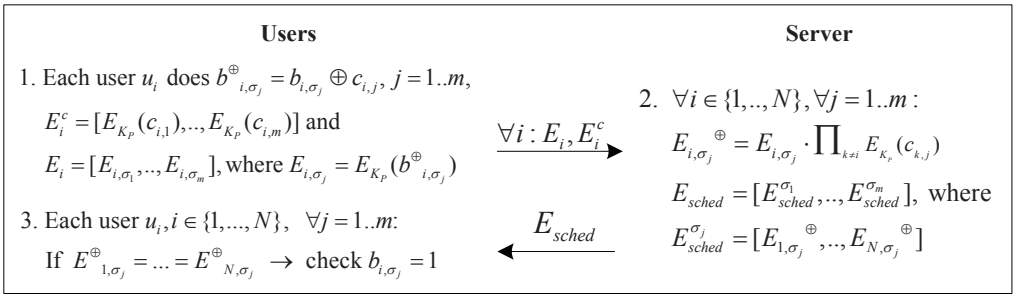


Figure 2.6: *SchedGM* protocol.

masks of all other users $u_k, \forall k \neq i$, obtaining $E^{\oplus}_{i,\sigma_j} = E_{i,\sigma_j} \cdot \prod_{k \neq i} E_{KP}(c_{k,j})$, $\forall i \in 1,\ldots,N$ and $\forall j = 1,\ldots,m$. Afterwards, the server sends all individual schedules, masked by a global mask $c_{1,j} \oplus \ldots \oplus c_{N,j}$, to each user in a random order. As a result, a user will not know his own schedule (masked with the global mask), otherwise he would be able to determine the global mask. Finally, each user decrypts the received messages and compares all masked individual schedules. If for a given time slot σ_j they all have the same value, then each user u_i can infer whether the time slot σ_j is available by looking at its own schedule b_{i,σ_j}. The following result shows the correctness and privacy properties of *SchedGM*.

Lemma 2.3. *The protocol* SchedGM *is correct and server-private.*

Proof. **Correctness** From Section 2.2.3, we know that any scheduling algorithm should output $f(Y)$, on inputs $f_1, f_2, \ldots, f_N$, where $f_i = f(b_{1,1}, \ldots, b_{i,m})$, such that each user is able to privately compute $Y = f^{-1}(f(Y))$, where $Y = y^1, \ldots, y^j, \ldots, y^m$. The output bit $y^j, \forall j$ should be such that it should take (with overwhelming probability) some value v if and only if all users are available. Otherwise, the output bit y^j never takes value v and should take some other value, indicating that at least one user is not available. In the case of *SchedGM*, each $f(y^j)$ (output by the server) consists of N different bits, one for each user, where each bit is the corresponding $b_{i,j}$ (schedule bit j of user u_i) masked by a global mask. From Figure 2.6, we can see that $y^j =$ "YES", for a particular user u_i, if and only if all of the N bits in $f(y^j)$ are equal and $b_{i,j} = 1$ (user u_i is available), and $y^j =$ "NO" otherwise. It is straightforward to see that all N bits in $f(y^j)$ will be equal only in two cases: 1) $b_{i,j} = 1, \forall i$ (all users are available) or 2) $b_{i,j} = 0, \forall i$ (all users are not available). Thus, $y^j =$ "YES" if and only if all users are available and $y^j =$ "NO" for any other case. Thus, *SchedGM* is correct.

Privacy

Hereafter we present the privacy proofs, both for user- and server-privacy.

- User identifiability advantage
  As for the previous two algorithms, the identifiability advantage of any user u_i for the *SchedGM* protocol is defined as

$$Adv_{u_i}^{IDT}(SchedGM) = \left| Pr_{u_i}[b'_{j,p} = b_{j,p} | r > 1, b_{i,p}] - \frac{1}{2} \right|$$

  where $1 \leq r \leq \lfloor N/2 \rfloor$ is the number of output elements that have the same value. Note that in *SchedGM* each user gets N masked output values $b_{i,p}^{\oplus}$, $\forall i \in \{1, \ldots, N\}$, for each time slot $p \in \{1, \ldots, m\}$, but it cannot unmask them as it does not possess the global mask. Therefore, any user knows that there are r masked bit values of one kind and $N - r$ of the other kind, without knowing whether one or the other kind corresponds to $b_{i,p} = 1$. Without loss of generality, we assume that the Challenger chooses user u_1 as the Adversary and we focus on the non-trivial case $N > 2$. By expanding the first term, we have

$$Pr_{u_1}[b'_{j,p} = b_{j,p} | r > 1, b_{1,p}] = \sum_{k=0}^{1} Pr(b'_{j,p} = b_{j,p} | r > 1, b_{1,p} = k) \cdot Pr(b_{1,p} = k | r > 1)$$

$$= \sum_{k=0}^{1} \sum_{z=0}^{1} Pr(b'_{j,p} = z | r > 1, b_{1,p} = k) \cdot Pr(b_{j,p} = z | r > 1, b_{1,p} = k) \cdot Pr(b_{1,p} = k | r > 1)$$

  From the above, we obtain

$$Pr(b'_{j,p} = 0 | r > 1, b_{1,p} = 0) = \frac{1}{2} \cdot \frac{C_r^N \cdot r}{C_r^N \cdot N} + \frac{1}{2} \cdot \frac{C_{N-r}^N \cdot (N - r)}{C_r^N \cdot N} = \frac{1}{2}$$

$$Pr(b'_{j,p} = 1 | r > 1, b_{1,p} = 0) = \frac{1}{2}$$

$$Pr(b'_{j,p} = 0 | r > 1, b_{1,p} = 1) = Pr(b'_{j,p} = 1 | r > 1, b_{1,p} = 1) = \frac{1}{2}$$

  which implies

$$Pr_{u_i}[b'_{j,p} = b_{j,p} | r > 1, b_{i,p}] = \frac{1}{2}$$

  and thus the final result

$$Adv_{u_i}^{IDT}(SchedGM) = 0, \qquad \forall N > 2$$

- User linkability advantage
  Hereafter we intuitively show that $\exists N > 2 | Adv_{u_i}^{LNK}(SchedGM) \geq 1/p(N)$, where $p(N)$ is any positive polynomial function of N. After Step 4 of the challenge-response game, the Adversary u_1 knows (i) its own schedule bit $b_{1,p}$ and (ii) the number r of masked schedules of one particular kind. Even though u_1 cannot determine with certainty whether the r elements

Table 2.2: Client and server specifications.

	Client (Nokia N810)	***Server***
Processor	TI OMAP 2420, 400 MHz	Intel Centrino Duo T2500, 2 x 2.00 GHz
RAM	DDR RAM 128 MB	DDR2 RAM 3 GB
OS	Maemo Linux OS2008 (Diablo)	Ubuntu 9.10, kernel 2.6.31.22

correspond to the "available" or to the "busy" state, it knows that the challenger picks the two other bits $b_{h,p}, b_{j,p}$, $j \neq h \neq i$, at random and therefore it also knows that the lower the value r, the greater the probability that any two bits in the sequence under consideration have the same value. Intuitively, if $r = 1$ it means that there are $N - 1$ schedules of one kind and only one schedule of the other kind. Therefore, the probability that any two users have same schedule value is greater than, for instance, when $r = \lfloor N/2 \rfloor$. Thus, the linkability advantage $Adv_{u_i}^{LNK}(SchedGM)$ is not less than $1/p(N)$, $\forall N > 2$, as $\exists r \in \{1, \ldots, \lfloor N/2 \rfloor\} | Adv_{u_i}^{LNK}(SchedGM) \geq 1/p(N)$ for some positive polynomial $p(N)$.

- Server advantages
  As in *SchedElG* and *SchedPa*, the server performing the *SchedGM* algorithm does not have access to any schedule bit. Therefore, *SchedGM* is server-private. ■

□

2.5 Implementation and Performance Evaluation

In this section we present the system and implementation details related to our three privacy-preserving scheduling algorithms. First, we describe the details about the systems and platforms on which we developed and implemented our applications. Second, we present the experimental measurements of the performance of our applications (both on the client devices and on the server), and we thoroughly discuss these results and compare the efficiency of all the algorithms.

2.5.1 Systems and Platforms

Clients and server systems The client application was run, tested and evaluated on the Nokia N810 devices. The server application was implemented and evaluated on a laptop. The hardware and OS specifications are listed in Table 2.2

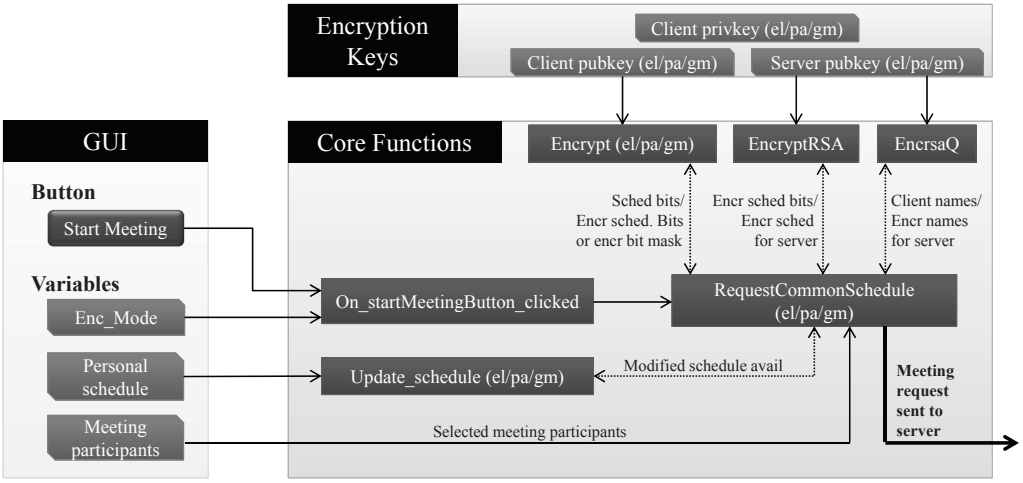


Figure 2.7: Flowchart showing the initiation of a meeting scheduling request on the client application. The function names (such as `encrypt (elg/pa/gm)` or `requestCommonSchedule (elg/pa/gm)` that appear in this figure are intuitive placeholders for the actual function names that are used in the client application.

Code specifications Our privacy-preserving scheduling applications were developed with the Qt 4.0 framework [119], using QtCreator as the IDE. The client application was ported to the N810 devices using the Maemo SDK on the *Scratchbox* cross-compilation toolkit[2].

Cryptographic libraries The *libgcrypt* standard GNU library[3] was used to implement the Elgamal and the RSA cryptosystems. Similarly, the *libpaillier* library[4] was used to implement the Paillier cryptosystem. For the Goldwasser-Micali cryptosystem, we did not find any existing available libraries, and therefore we developed a new library, *libgm*, to implement the basic cryptographic operations. We intend to release our *libgm* library to the public under the GPL licence.

2.5.2 Software Architecture

Our privacy-preserving activity scheduling software consists of two applications: the *client* and the *server*. The client application runs on the Nokia N810 mobile device, and has a GUI to take inputs from the users. The server application runs on the Intel-based PC and is managed through the standard Unix console.

[2]Details on the Scratchbox and Maemo SDK are available at http://maemo.org/maemo_release_documentation/maemo4.1.x/node4.html.

[3]The documentation for *libgcrypt* is available at http://www.gnupg.org/documentation/manuals/gcrypt/index.html

[4]Source code available at http://acsc.cs.utexas.edu/libpaillier/.

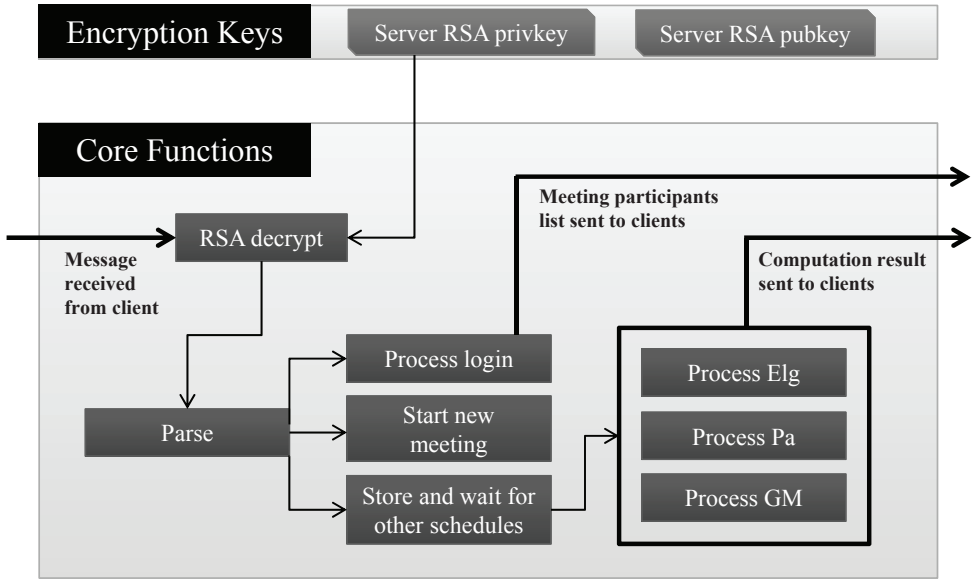


Figure 2.8: Flowchart showing the server application structure when handling meeting requests and replies from and to the clients. The function names (such as `RSA Decrypt` or `Process Elg`) that appear in this figure are intuitive placeholders for the actual function names that are used in the server application.

Client Application

The client application stores the schedules of the users and displays the list of potential meeting participants for each user. This list is maintained and managed by the user himself, who can choose the meeting participants before initiating the meeting scheduling procedure. Each user can use the GUI to set his availabilities, send a meeting scheduling request, reply to an ongoing meeting request or refuse to participate in a received meeting request. To send a meeting scheduling request, the initiator first selects one of the available privacy-preserving algorithms (*SchedElG, SchedPa* or *SchedGM*) and the intended meeting participants. Then, the procedure is initiated by a click on the "Start meeting" button. Figure 2.7 shows a flowchart of the application on the client device, when a user sends a request to schedule a meeting.

Server Application

The server is a GUI-less application that interacts with the clients to handle requests such as login and computation of common availabilities. The main server class, *ScServer*, inherits *QTcpServer* and is used as the server socket. Figure 2.8 shows the server flowchart structure.

More details about the inner structure of the server will be made available to

Table 2.3: Efficiency and privacy comparison with the scheduling protocols DisCSP [158], MPC-DisCSP2 [143] and SDC [79]

		Per-user encr.	*Per-user decr.*	*Per-user comm.*	*Order of an encr. availab.*	*Privacy properties*
Centralized	***SchedElG***	O(m)	O(m)	O(m)	1024 bits	User-private Server-private
	SchedPa	O(m)	O(m)	O(m)	2048 bits	User-private Server-private
	SchedGM	O(m)	O(N · m)	O(N · m)	1024 bits	User-private [#] Server-private
	Naïve	0	0	O(m)	1 bit *	None
Hybrid	**DisCSP protocol**	O(m)	O(m)	O(N · m)	1024 bits	Private
Distributed	**MPC-DisCSP2 protocol**	O(N · m)	O(m)	O(N · m)	2048 bits	Private
	SDC protocol	$O(N^2 \cdot m)$	O(N · m)	$O(N \cdot m \cdot \lceil \log_2(N) \rceil)$	1024 bits	Private

([*]) The naïve algorithm does not encrypt the schedule bits
([#]) Adv^{IDT} is a negligible function, whereas, for some output Y of the algorithm, Adv^{LNK} is non-negligible

the public, together with the source code, under the GPL licence.

2.5.3 Experimental Performance Evaluation

Before presenting the performance measurement details, let us first perform a comparative analysis of the asymptotic complexities of the proposed protocols, as shown in Table 2.3. In order to compare our three algorithms with an equivalent security, we set the bit-lengths of the ElGamal modulus q and the Paillier and GM modulus n to 1024 bits. A time-slot availability would then be encrypted to a 2-tuple of 1024-bit ciphertexts for ElGamal, to a 1024-bit ciphertext for GM and to a 2048-bit ciphertext for the Paillier encryption scheme.

From Table 2.3 we can see that the *SchedElG* and *SchedPa* protocols are very efficient, both in terms of communication $O(m)$, where m is the number of time slots, and computation complexity $O(m)$. Moreover, these two algorithms provide strong privacy guarantees. *SchedGM*, on the contrary, is comparatively less efficient due to the greater number of exchanged messages ($O(N \cdot m)$, where N is the number of participants). From the privacy perspective, *SchedGM* reveals more information: users can infer the ratio of free/busy participants for each time slot without identifying those that are busy and those that are free. Because

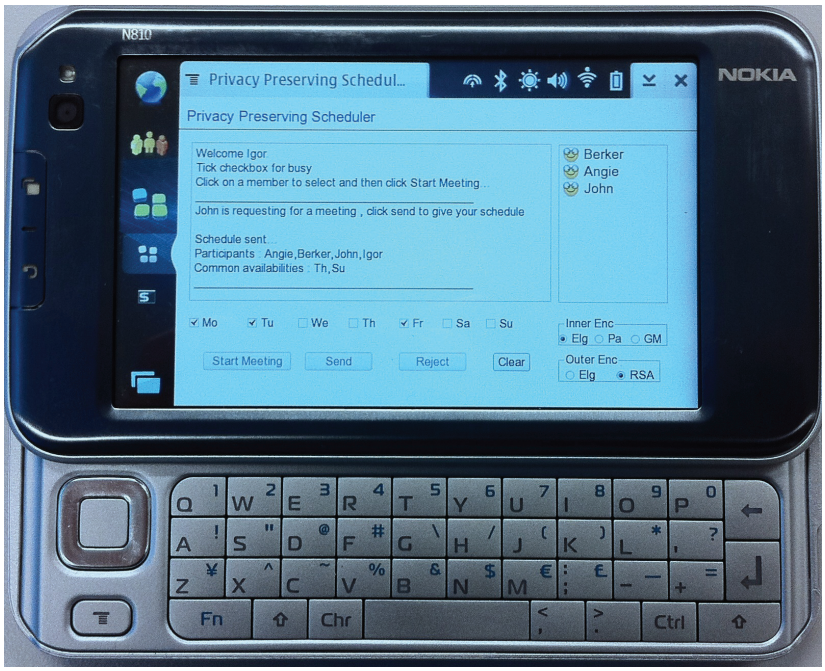


Figure 2.9: Frontend of the scheduling application on a Nokia N810.

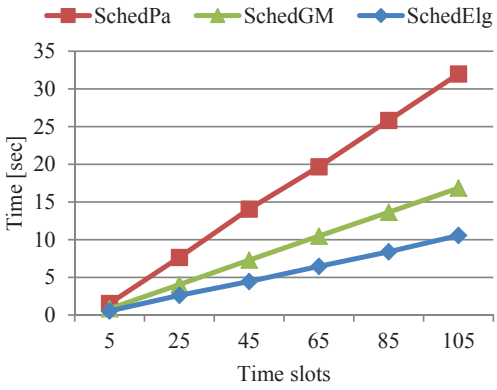
in all schemes, the server operates only on encrypted data, it cannot gain any knowledge about the users' private schedules.

Distributed [143, 79] and hybrid [158] solutions proposed in the literature are less efficient from the communication standpoint as compared to the proposed protocols. Moreover, the computational complexity of these schemes is higher than *SchedElG* and *SchedPa*, and this undermines their applicability on resource-constrained mobile platforms. Even though the hybrid approach [158] has comparable computation complexity, it is not completely reliable from the privacy point of view because it assumes that the server(s) can get clear-text access to the individual availabilities.

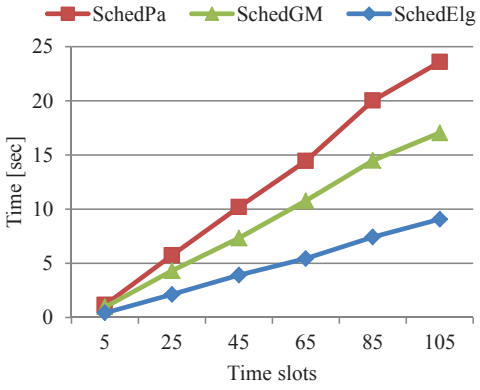
We further evaluate the performance of *SchedElg*, *SchedPa* and *SchedGM* by implementing the client component of the protocols and primitives on Nokia N810 mobile devices with a 400 MHz CPU and 128 MB RAM (Figure 2.9), and the server component on a desktop computer with a 2 GHz CPU and 3 GB RAM. The results of the experimentation are shown in Figure 2.10.

Client Encryption

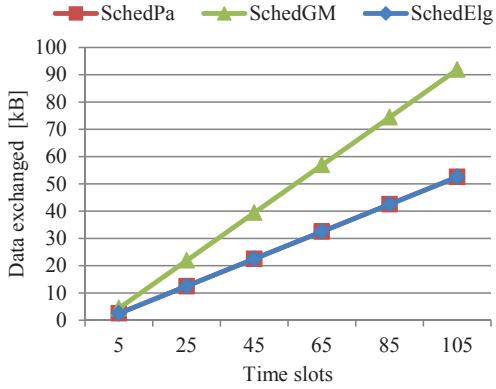
As we can see from Figure 2.10, the time required to perform the scheduling operations increases with the number of time slots for all the proposed algorithms, which is intuitive. With respect to encryption performance, Figure 2.10(a) shows that *SchedElg* is the most efficient scheduling algorithm, requiring 4 seconds to



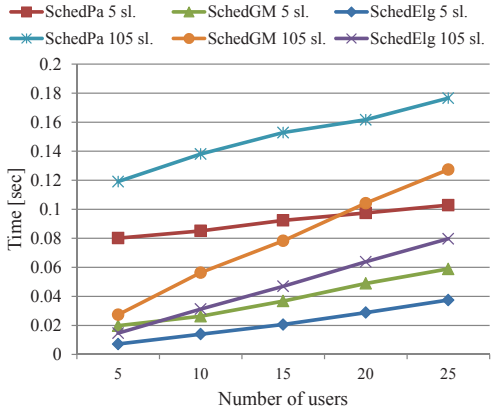
(a) Client encryption performance



(b) Client decryption performance. We considered $N = 5$ for *SchedGM*.



(c) Client communication efficiency (at application layer). We considered $N = 5$ for *SchedGM*.



(d) Server processing performance.

Figure 2.10: Testbeb implementation performance measurements.

encrypt 45 time slots (a typical weekly schedule on a per hour basis). The same task is accomplished by *SchedGM* and *SchedPa*, respectively, in 7 and 14 seconds. These results might be explained by the following. First, the cryptographic primitives for the ElGamal scheme are implemented in a standard well-optimized library, *libgcrypt*, present in most Unix-based operating systems. *SchedGM*, on the contrary, does not use a standard library and can be further optimized. Second, the encrypted elements in *SchedPa* have twice the bit-length of those used in the other two algorithms, and therefore the same operations (multiplications and exponentiations) require more time.

Client Decryption

Figure 2.10(b) shows the time required for decrypting the final result (common availabilities) of the scheduling algorithms at the client. Similarly to the encryption time, the fastest algorithm for the decryption is *SchedElg*, which takes 4 seconds in order to obtain the aggregated availabilities for a 45 time-slot period. For the same number of time slots, *SchedPa* takes approx. 7 seconds, which is almost twice longer than the best performance. The decryption times for both *SchedElg* and *SchedPa* are independent of the number of participants. The performance of *SchedGM*, due to the fact that the final output of the algorithm is a sequence of vectors instead of just a single aggregated vector, decreases with the number of users, as well as with the number of time slots. Thus, for a reasonable number of participants (e.g. $N = 5$), *SchedGM* is still practical enough to be implemented on resource-constrained mobile devices, although it is not the preferred solution.

Client Communication

Figure 2.10(c) shows the (application layer) data that each client exchanges during one execution of the scheduling algorithm. In general, all the proposed privacy-preserving scheduling algorithms have reasonable communication costs. *SchedElg* and *SchedPa* are the most efficient algorithms and they require 22 kB of data in order to compute the aggregated availabilities of a 45 time-slot period, whereas *SchedGM* requires 39 kB for the same result. As previously mentioned, *SchedGM* uses a sequence of masked vectors in order to compute the final availabilities of the users, and therefore the amount of data is proportional both to the number of users and time-slots.

Server Performance

The scheduling server's performance is shown in Figure 2.10(d). As it can be seen, the time required to perform the scheduling operations on encrypted values increases with both the number of users and time slots. For instance, the running time (in seconds) for the server implementation of the *SchedElG* algorithm is at most $2 \cdot N \cdot m \cdot T_{mul-ElG}$, where N is the number of clients, m the number of time-slots and $t_{mul-ElG}$ is the time required to compute one multiplication operation between two $\lceil \log(q) \rceil$-bit integers (q is the order of the group in the ElGamal encryption scheme). The running time for the *SchedPa* and *SchedGM* is, respectively, at most $N \cdot m \cdot T_{mul-Pai} + m \cdot T_{exp-Pai}$ (where $T_{mul-Pai}$ and $T_{exp-Pai}$ is the time required to perform a multiplication and an exponentiation respectively of two $\lceil \log(n^2) \rceil$-bit integers) and $3 \cdot N \cdot m \cdot T_{mul-GM}$ (where T_{mul-GM} is the time required to perform a multiplication between two $\lceil \log(n) \rceil$-bit integers).

As it can be seen, even with a large number of users and time slots, the amount of time required for the server-side scheduling operations is still below 0.2 seconds, which suggests that the load on the server is limited, which allows it to efficiently handle multiple scheduling events, without incurring in huge computational overhead.

2.6 User Study

In this section we present the modalities and results of the user study that we carried out with our prototype meeting-scheduling application. The goal of this study was to assess the sensitivity of the subjects to privacy issues in meeting-scheduling applications, as well as to obtain feedback with respect to our prototype application.

2.6.1 Background

Based on the privacy- and usability-related questionnaire guidelines from ([33, 98]), we prepared and conducted a targeted user-study on 19 subjects, sampling a population of university students (both undergraduate and graduate), non-scientific personnel and people from a non-technical environment.

The entire study was divided into three phases, with two different sets of questions that were given in *Phase 1* and *Phase 3* respectively. In *Phase 1*, the participants were asked to reply to a set of 20 questions before using the meeting scheduling application. In *Phase 2*, they were asked to use our prototype application to schedule meetings with the other participants both in a controlled and uncontrolled setting; the first time, we instructed them how to use the application, and afterwards they were free to use it as they pleased. Finally, in *Phase 3* the participants answered a second set of 14 post-experience questions, after having used our prototype application.

The goal of Phase 1 of the study was to assess the participants' level of adoption of mobile technology and applications, and to get their opinion on privacy issues in such applications. The participants were not told beforehand what kind of mobile application they will be asked to use in Phase 2. During Phase 1, the respondents answered the *Pre-Experience A* questionnaire, which comprises 20 questions on both generic technology topics (such as usage and ownership of mobile devices, utilization of mobile social networks and calendar/agenda) and more specific privacy-related questions (such as their online behavior and opinions on information release). For instance, one statement related to users' online behavior and privacy is "I am willing to use my real name in online discussions (forums, chat rooms, etc.)", to which the respondents had to answer with either *Disagree*, *Tend to disagree*, *Tend to agree* or *Agree*.

After Phase 1 was completed, in Phase 2 we instructed the participants on the specifics of our prototype scheduling application and how it works, in a step-by-step fashion. We then asked them to execute one instance of the scheduling process. Next, we told them to use the application as they please, without the experimenters overseeing the process. The goal of Phase 2 was to show our application and to let the participants use it autonomously, in order to get an opinion for the Phase 3 of the study.

The goal of Phase 3 was to obtain feedback on different performance and privacy aspects of our prototype application. The respondents answered the *Post-Experience B* questionnaire, which comprises 14 questions centered on our application prototype, its perceived usefulness, efficiency, ease of use, and privacy. For example, the statement "I could easily identify who was/were the person/people that were not available for a particular time slot" could be answered by *Disagree*, *Tend to disagree*, *Tend to agree* or *Agree*.

Hereafter we provide the summary of the results and discussion on our user-study.

2.6.2 Results

Phase 1

Technology Utilization In this first part, we discuss the results concerning the technology utilization habits of the respondents. With respect to mobile applications, our results show that 63% the respondents browse the Internet with a mobile device, whereas 53% of them use the mobile calendar/agenda application on their devices in order to organize meetings. 86% of such meetings are scheduled once or twice a week, and most of the time (89%) such meetings involve 2-4 people. In order to reach a consensus, the meeting participants use e-mail 58% of the time and the telephone for the remaining 42%. Social networks, such as Facebook or Twitter, are used by 84% of the respondents, and 44% of them access such services using their mobile devices. These results suggest that although meeting scheduling and calendar management using mobile devices is already a reality, people still struggle to reach a consensus in an efficient way. In order to agree on a common time slot by using e-mail, multiple rounds of interaction among the meeting participants are required.

Privacy Attitudes In this second part, we discuss the privacy concerns of the respondents when using everyday applications. In general, 63% of the respondents tend to disagree or disagree with the statement "I would put photos/videos of myself, my family and friends on the Internet". When asked about third parties sharing personal information about them, 89% of the respondents agree that no third party should disseminate users' private information without their knowledge. With respect to privacy in online interactions, 63% feel that they

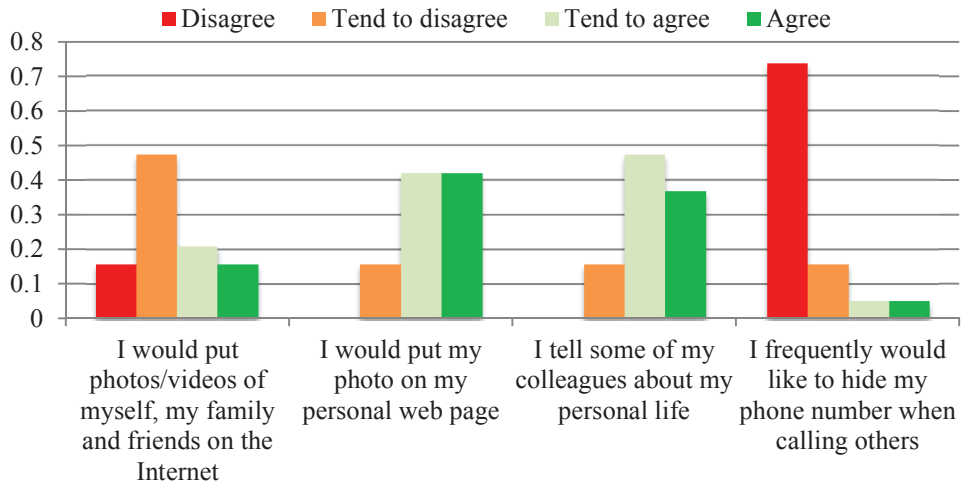


Figure 2.11: Extract of the user-study questionnaire about people's privacy attitudes.

would prefer not to use their real name and use pseudonyms instead. Figure 2.11 shows other interesting privacy attitude results. In summary, our respondents tend to by sensitive to the privacy issues related to the use of mobile applications, and thus effectively controlling the access to and dissemination of personal information is a valuable differentiator for mobile applications.

Scheduling Applications and Privacy The third part of the results show the opinion of respondents about meeting-scheduling applications on mobile devices. According to the results, 84% of the respondents are not aware of any existing mobile application for meeting scheduling. Among those, 43% would be quite (or a lot) interested in having such applications. With respect to privacy, 58% would be comfortable in sharing their basic schedule availabilities with the other meeting participants, while none of them would be willing to share all the details (such as place, time and subject) about these availabilities.

With respect to priorities in mobile meeting-scheduling applications, Figure 2.12 shows the choices of the respondents, ordered by the perceived priority (on a scale from 1 to 4, where 1 is the top priority and 4 is the least priority). The figure shows that privacy is perceived as the first priority in mobile meeting-scheduling applications 33% of the time. If we consider the cumulative result for the 1st and 2nd priorities, privacy achieves a total of 77%. Although the ease of use of the application is perceived as the top priority for 50% of the respondents, the cumulative result for the 1st and 2nd priorities achieves 67%, which is 10% less than privacy. The speed and the Graphical User Interface (GUI) have the least priority for the users, where speed is only the third priority most of the time, and the GUI is almost exclusively the least priority.

Overall, the results suggest that privacy is indeed perceived as being the top

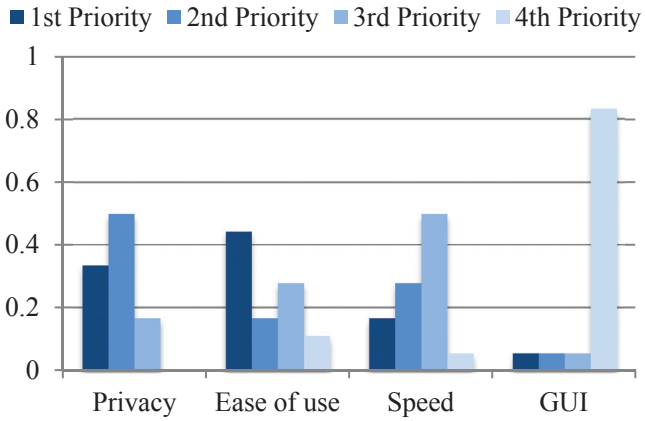


Figure 2.12: Extract of the user-study questionnaire about people's priorities in mobile scheduling applications.

or the second priority in meeting-scheduling applications, which is in line with the concerns that the respondents had before using our application. From a software developer standpoint, this means that both ease of use and privacy need to be taken into account from the beginning of the application development process. In particular, the privacy mechanisms should be implemented in a way that does not significantly affect the usability or performance. The acceptance of meeting-scheduling applications is thus highly influenced by the availability of effective and intuitive means for controlling privacy preferences.

Phase 3

User-Experience Figure 2.13 shows some interesting results about the perceived user experience while using our prototype meeting-scheduling application on the Nokia N810 devices. As it can be seen, almost 70% of the respondents agree that they were able to perform the meeting scheduling task quickly by using our application. Moreover, 95% of them agree that it was easy for them to learn to use our application. Regarding the information presented on the screen, users mostly agree that it was easy to find all necessary information, such as the meeting participants, the individual schedule and control buttons. Similar results have been obtained for the organization of the user interface.

These results suggest that it is indeed possible to integrate simple privacy mechanisms into mobile application, without incurring in significant learning overhead. A clean GUI with a transparent integration of privacy features proved to be very effective in this regard.

Privacy in Our Prototype Application In this last part, we discuss the subject of privacy with respect to our prototype application, and how its implementation was perceived by the respondents. Figure 2.14 shows some of the

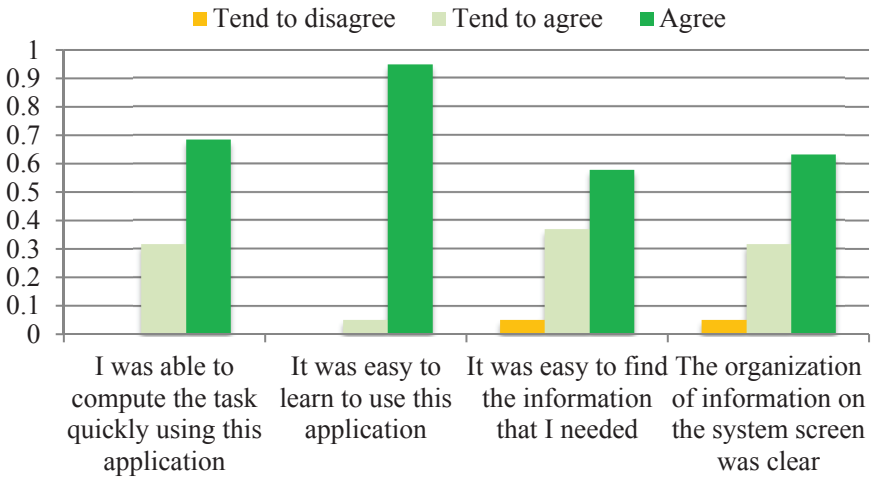


Figure 2.13: Extract of the user-study questionnaire about the user experience for our prototype application.

results obtained from the user study. In general, all respondents tend to agree or agree that it is important to not reveal any more information to the central server than strictly necessary. When asked about the way privacy has been implemented in our prototype application, 95% of them claim that they could not identify the people who were available (or not) in a given time-slot. Concerning the potential overhead due to the privacy mechanisms, 71% of the users feel that having the privacy feature in such application did not make it more complicated for them to use it; only 5% tend to agree with the opposite.

Regarding the third-party knowledge of the individual schedules, 74% agree that they felt comfortable knowing that the central scheduling server did not know their private schedules, and only 5% of them disagree. The users were told about this feature during Phase 2 of the study. However, when the third-party is the other meeting participants (and not the central server), 47% felt comfortable knowing that their privacy was preserved. Nevertheless, this percentage increases to 95% when considering responders who tend to agree with such statement, in addition to those who agree.

In summary, this user-study has shown that the majority of the respondents are concerned about their privacy in scheduling applications, and that they would welcome effective and simple means for protecting it and still enjoy such services. Our prototype application has proven to be effective in both providing a user-friendly interface for the meeting scheduling participants, and a transparent way to ensure that privacy of individual schedules is preserved. The results have also shown that there is no significant overhead for using privacy in such applications, and that people appreciated having the ability to not disclose more information

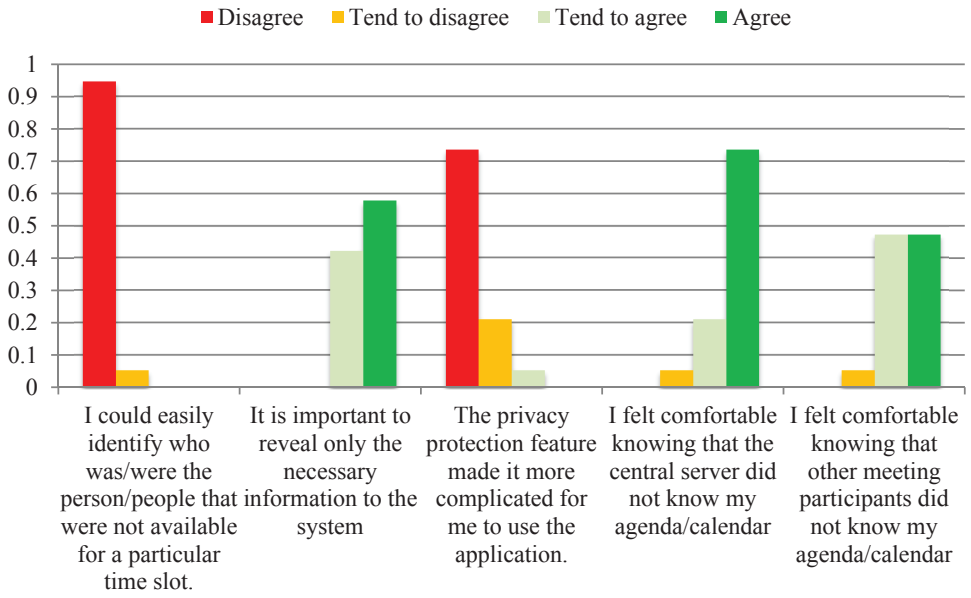


Figure 2.14: Extract of the user-study questionnaire about people's opinions on the privacy features in our prototype application.

about their schedules than what was strictly necessary in order to compute the available time slots.

2.7 Extensions

In this section, we show how *SchedPa* can be easily extended to the case where user schedules are non-binary, i.e., each time slot is a non-negative cost $C_{i,j}$ that indicates u_i's preference for time-slot j. We also describe several active attacks on the proposed scheduling schemes, such as collusion between users-server and data modifications by the users, and how these attacks can be mitigated by using existing cryptographic mechanisms. Finally, we discuss some further enhancements for the privacy of users' schedules and how to implement them.

2.7.1 Non-Binary Schedules

The goal here is to find, in a privacy-preserving fashion, the time-slot with the minimum aggregated cost. The scheme works as follows:

1. Each user u_i reorders his cost sequence $C_{i,1} \dots C_{i,m}$ using the shared permutation σ and encrypts each cost C_{i,σ_j} in the sequence using the Paillier cryptosystem with the shared group key K_P. He then passes the result $(E_{K_P,r_{i,1}}(C_{i,\sigma_1}) \dots E_{K_P,r_{i,m}}(C_{i,\sigma_m}))$ to the server.

2. The server computes the encrypted sum of costs $E_{K_P,r_j}(R \cdot \sum_{i=1}^{N} C_{i,\sigma_j}), \forall j$, where R is a random integer (greater than one) chosen by the server.

3. The server selects a pre-determined user u_k and passes a *randomly ordered* (different from σ) sequence of the encrypted aggregated costs to it. This is to prevent u_k from learning the aggregated cost function.

4. User u_k decrypts all the elements passed from the server, and identifies the minimum aggregated cost.

5. User u_k then queries the server for the index of the (encrypted) minimum aggregated cost. The server then distributes the queried index to all users.

It can be easily shown that the above scheme is execution privacy-preserving.

2.7.2 Active Attacks

There are five kinds of possible active attacks on the scheduling schemes: (i) collusion between the scheduling server and users, (ii) collusion among users, (iii) fake user generation by the server, (iv) individual user schedule modification and (v) integrity and replay attacks.

In order to thwart the first issue, the invited participants could agree to establish a shared secret by using techniques from threshold cryptography, such as [146]. The server should then collude with at least a predefined number of participants in order to obtain the shared secret and learn the individual availabilities. The second concern may arise if k colluding users set their schedules to *all-available*, and try to learn the schedules of other users. Assuming that N is the total number of participants and k the number of colluding ones, our schemes would provide some level of schedule privacy to honest users, as long as $N - k \geq 2$. Only if all but one users collude, then they would be able to determine the schedule of the remaining user. In order for the third attack to succeed, the server would need to generate fake users and convince the true participants about the legitimacy of the fake users. In practice, this is a non-trivial task to achieve, and thus the attack has a very slim chance of succeeding. Moreover, the effectiveness of such an attack could be further reduced by adopting the threshold cryptographic scheme mentioned previously, because the server would then need to generate k fake users and validate them as true participants.

The fourth attack is also not able to succeed in revealing the availability of other meeting participants, as the best a malicious user can do is to set its own schedule to all-available, and then guess the availabilities of the other $N - 1$ participants. Even if a malicious user attempts to modify its own schedule with invalid values, such as negative values, the message domain restrictions of cryptosystems (such as ElGamal and Paillier) would prevent such modifications.

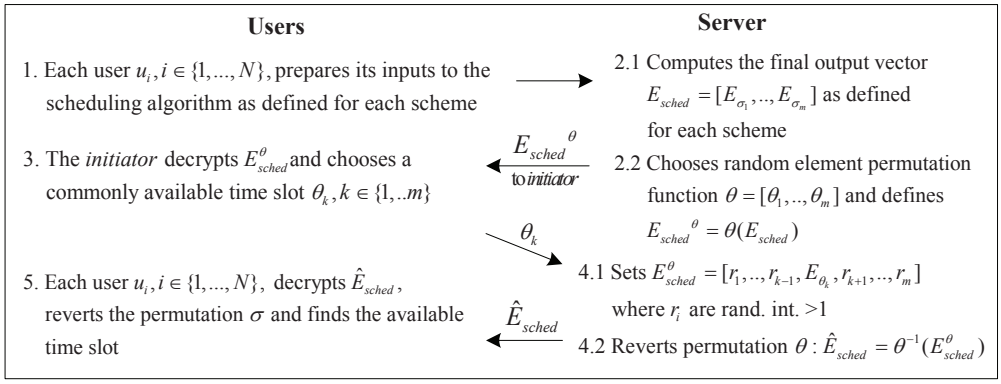


Figure 2.15: Extended algorithm scheme for revealing a single available time slot.

Thus, malicious attacks consisting of manipulating the final result by using invalid negative values as schedule values are not possible in the proposed protocols.

The last attack concerns the integrity and freshness of the encrypted schedules. The participants are the only entities in the system that know the secret that has been used to generate the public/private key pair, and therefore they are the only ones that can generate and verify the integrity of the encrypted data. Moreover, using the shared common secret, each participant could generate a fresh *nonce* at each algorithm execution and send it (in encrypted form) to the server during the scheduling process. The server would then forward these encrypted nonces to each participant, who could verify that all received nonces are equal. If not all nonces are equal, then the participants know that there has been at least one replay attack, and thus the schedule results are not to be trusted.

2.7.3 Single Available Time Slot

The output of conventional, non-privacy-preserving scheduling services (such as Doodle [45] or Outlook [110]) consists of time slots in which all participating users are available. The proposed schemes follow this paradigm and they provide, in an efficient and privacy-preserving way, all time-slots for which all users are available.

In some cases, however, it might be desirable to limit the disclosure of common availabilities to only one time-slot, instead of the set of all available time-slots. This would provide an additional layer of privacy for the individual schedules, as the participants would be given a single feasible solution. Hereafter we describe one simple way to adapt the proposed schemes to support this feature (Figure 2.15).

First, all users participating in the scheduling process perform Step 1 of the respective algorithm (*SchedElg*, *SchedPa* or *SchedGM*). Second, the server performs Step 2 but it does not send the final output to each user. Instead, it randomly chooses a private time-slot permutation function $\theta = [\theta_1, \ldots, \theta_m]$ and

applies it to the elements of the final output vector(s) E_{sched}. We call this new vector(s) E^{θ}_{sched}. At this point, the schedules have been permuted twice, once by the users prior to the encryptions (with σ) and once by the server (with θ).

Next, the server sends E^{θ}_{sched} to the user who started the activity scheduling (the *initiator*), which then gets the common availabilities but in a doubly permuted order. The initiator is able to determine the *available* slots in this doubly permuted time slot list, but he is not able to determine the time slots they correspond to in the original schedule. The initiator selects one commonly available time slot θ_k and securely sends the index θ_k to the server. Fourth, the server (i) replaces all availabilities other than θ_k in E^{θ}_{sched} with random numbers, (ii) reverts the permutation θ, and (iii) sends this new vector(s) $\hat{E}_{sched}$ to each user. Finally, each user decrypts and reverts the initial permutation σ of the received vector(s) and determines which time slot j is the only commonly available time slot.

This simple solution that reveals only a single available time slot to all the participants involves one extra message exchange between the initiator and the scheduling server, as shown in Step 3 of Figure 2.15. Although the permutation θ performed by the server preempts the initiator from knowing the true common availabilities, he might still want to maliciously modify the permuted availabilities. However, the only action the initiator can do is to choose one of the permuted time slots and communicate its index θ_k to the server, as it is the server who will then revert the permutation θ and send the final vector(s) $\hat{E}_{sched}$ to all users.

2.8 Related Work

In the literature, the four most relevant bodies of work that address privacy in scheduling or similar scenarios are based on techniques from private set-intersection [93, 41], distributed constraint satisfaction [155, 158, 143, 142], secure multi-party computation [79, 46] and e-voting [92]. Hereafter, we review the most relevant aspects of such approaches.

In the private set-intersection domain, Kissner and Song [93] use mathematic properties of polynomials to design privacy-preserving union, intersection and element reduction operations on private multisets by leveraging on the Goldwasser-Micali homomorphic encryption scheme [69]. De Cristofaro and Tsudik [41] provide efficient variations of private-set intersection protocols and present a comparison in terms of computational and communication complexity, adversarial model and privacy. The authors also give informal definitions of client and server privacy. However, PSI approaches are generally distributed, and an efficient extension to an n-party protocol is challenging. In the meeting scheduling scenario, for instance, a trivial extension of the 2-party PSI to n parties (by running a 2-

party protocol between each pair of users) would undermine the privacy of users' schedules as well; knowing the personal availability and the aggregate availability is sufficient to infer the other party's schedule.

Distributed constraint satisfaction approaches were investigated by Wallace and Freuder [155]: they study the tradeoff between privacy and efficiency and show that the information that entities learn during the negotiation of a common schedule has, in some cases, a tremendous impact on privacy. Details of an accept/reject response are exploited by intelligent agents in order to successfully infer the availabilities of other peers involved in the scheduling process. Similarly, Zunino and Campo [163] design a scheduling system in which entities learn and refine their knowledge about user preferences by using a Bayesian network. Yokoo *et al.* [158] use secret sharing among third-party servers in order to determine a suitable agreement among entities in a collusion-resistant way.

Solutions based on secure multi-party computation were investigated in [46] and a practical scheme was proposed in [79]. Herlea *et al.* [79], for instance, design and evaluate a distributed secure scheduling protocol by relying on properties of the XOR operation over binary values, in which all users contribute to the secrecy of individual schedules while ensuring the correctness of the results. Although not a pure e-voting scheme, Kellerman and Böhme [92] proposed an event scheduling protocol that inherits several security and privacy requirements from the e-voting context. However, a formal study of such properties and experimental performance results are missing in their work.

In contrast to most of the above solutions, we take a more centralized approach (with a single third-party server) for the privacy-preserving scheduling problem. Our solutions overcome communication and computational complexities intrinsic to most distributed approaches discussed above, as well as ensure that no private information (other than the resulting common availabilities) is exposed. Moreover, our protocols can easily fit into today's popular provider-consumer service architectures without incurring a huge communication cost on the service-provider.

2.9 Summary

Activity-scheduling applications are increasingly used by people on-the-move in order to efficiently and securely manage their time. In addition to privacy, which is paramount, such services should also be practical and feasible to implement, given the client-server paradigm that most providers use, and they should be as transparent to the user as possible. In this chapter, we have provided a framework for the formal study of privacy properties in such applications, and we have proposed three novel privacy-preserving protocols that, in addition to guaranteeing privacy, are more efficient than similar solutions in terms of computation and com-

munication complexities. Our implementation and extensive performance evaluation on real mobile devices demonstrates that the proposed privacy-preserving schemes are well suited to practical network architectures and services. Moreover, a thorough user-study of the prototype application suggests that our algorithms and software architecture are seamlessly integrated with the privacy-preserving algorithms, in a way that does not impede the user from quickly and effectively utilizing our application.

Publications: [21, 18]

Chapter 3

Privacy-Preserving Location Determination

As mentioned in the previous chapter, today's highly-interconnected urban population is increasingly dependent on mobile gadgets to organize and plan their daily lives. In addition to popular schedule management applications on these devices, such as Doodle mobile and Tymelie, various other urban applications and services are also gaining significance. Taxi-sharing applications, route-planning and participatory sensing services are a few noticeable examples. These applications most often rely on current (or preferred) locations of individual users or a group of users, which jeopardizes their privacy; users do not necessarily want to reveal their current (or preferred) locations to the service provider or to other, possibly untrusted, users. In this chapter, we propose privacy-preserving algorithms for determining an optimal meeting location for a group of users. We perform a thorough privacy evaluation of the proposed approaches, under both passive and active adversarial models, by formally quantifying privacy-loss in this setting. In order to study the performance of the proposed approaches in a real deployment, we implement and test their execution efficiency on Nokia devices. By means of a targeted user-study, we attempt to get insight into the privacy-awareness of users in location-based services and the usability of the proposed solutions.

Chapter Outline In Section 3.1, we introduce the location determination problem. We present the system architecture in Section 3.2 and the problem definition in Section 3.3. In Section 3.4, we formalize the privacy requirements and present our algorithms. We then perform a comparative analysis and discuss the implementation results in Section 3.5, and we summarize the results of our user-study in Section 3.6. We discuss the extensions of our schemes in Section 3.7, and we present the related work in Section 3.8. We summarize this chapter in Section

3.9.

3.1 Introduction

Advances in handheld device and smartphone technology, coupled with their rapid proliferation in urban communities, have enabled users to be constantly in connection with each other and to enjoy context-rich services on their devices while on-the-move. Online service providers take advantage of this dynamic and ever-growing technology landscape by proposing novel and innovative context-based services or by enhancing existing ones for the end-users. These services heavily rely on the improved computing and context-sensing capabilities of modern-day mobile devices and smartphones. Location-based services (LBS), an example of one such type of context-based services, are used by millions of mobile subscribers every day in order to obtain location-specific information and services [52].

Two popular features of location-based services are *location check-ins* and *location sharing*. By checking into a location, users can share their current location with family and friends or obtain location-specific services from third-party providers or businesses [58]. The obtained service does not depend on the locations of other users. Facebook, for instance, recently launched a check-in service by which users can benefit from on-the-spot discounts and deals from localized businesses [51]. The other type of location-based services also becoming popular are those that rely on the sharing of locations (or location preferences) by a group. According to a recent study [111], location sharing services are used by almost 20% of all mobile phone users. One noticeable example of such a service is the taxi-sharing application, offered by a global telecom operator [123], where smartphone users can share a taxi with other users at a suitable location by revealing their departure and destination locations. Similarly, another popular service [97] enables a group of users to find the most geographically convenient place to meet by minimizing the distance everyone has to travel. Several other providers [108, 109] offer variants of this service, either as on-line or stand-alone applications.

One important, and often overlooked, concern in location-sharing-based applications is the privacy of the user's location information or location preferences vis-à-vis the service provider and other, potentially untrusted, users. For instance, such information can be used to de-anonymize users and their availabilities [70], to track their preferences [62] or to identify their social networks [61]. For example, in the taxi-sharing application, a curious third-party service provider could easily deduce the home/work location pairs of users who regularly use their service. Without effective protection, even sparse location information has been shown to provide reliable information about users' private spheres, which could

have severe consequences on their social, financial and private lives [95]. Malicious entities, if they get hold of this private information, could use it for all kinds of nefarious activities [129]. Even service providers who legitimately track users' location information in order to improve the offered service can inadvertently harm users' privacy, if the collected data is leaked in an unauthorized fashion or improperly shared with corporate partners. Thus it is clear that, in any location-sharing-based service (LSBS), the disclosure of private locations to untrusted third-parties and peers must be limited. Recent user studies [111] on location information sharing in LSBSs, including the one we present in this chapter, show that end-users express the same concern. Our study on 35 participants, including students and non-scientific staff, showed that a large percentage of users (nearly 88%) are extremely concerned about sharing their location information.

In this chapter, we study privacy issues in location sharing-based services and propose practical, efficient solutions for the same. In line with this goal, we focus on a specific problem in LSBS, called the *fair rendez-vous point (FRVP)*. Given a set of user locations or location preferences, the FRVP problem is to choose an optimal (or *fair*) location among the proposed ones such that the maximum distance between this location and all other users' locations is minimized. Our goal is to provide practical, privacy-preserving techniques to solve the FRVP problem, such that neither a third-party nor users can learn about other users' locations; participating users learn only the optimal location. The FRVP problem is not only an important prerequisite in most LSBSs but it also nicely captures the significant privacy threats that can arise because of location sharing in LSBSs.

In particular, we first analytically model the privacy requirements in the FRVP problem and propose two privacy-preserving solutions for the same. We further evaluate the robustness and resilience of our schemes against various passive and active attacks and provide an in-depth privacy analysis of the proposed solutions. By means of a prototype implementation on a testbed of Nokia mobile devices, we analyze the efficiency and performance of the proposed algorithms on real systems. We also extend our existing solutions and implementations [20] to include cases where users could provide more than one location preferences and could assign priorities to those preferences. Finally, by conducting a targeted study of participants who used and interacted with our prototype, we attempt to gain insight into the current level of privacy-awareness and privacy requirements of LSBS users and, at the same time, verify the feasibility and acceptance of the proposed solutions.

3.2 System Architecture

We consider a system composed of two main entities: (i) a set of users[1] (or mobile devices) $\mathbb{U} = \{u_1, \ldots, u_N\}$ and (ii) a third-party service provider, called *Location Determination Server (LDS)*. Each user's mobile device is assumed to be able to establish communication with the LDS, either in a P2P fashion or through a fixed infrastructure-based Internet connection. The mobile devices are able to perform public-key cryptographic operations, and each user u_i has means of determining the coordinates $L_i = (x_i, y_i) \in \mathbb{N}^2$ of his preferred rendez-vous location (or his own location) by using a common coordinate system. We consider a two-dimensional position coordinates system, but the proposed schemes can easily be extended to other practical coordinate systems. For instance, the definition of L_i can be made fully compliant with the UTM coordinate system [154], which is a plane coordinate system where points are represented as a 2-tuple of positive values (distances in meters from a given reference point).

We define the set of the preferred rendez-vous locations of all users as $\mathbb{L} = \{L_i\}_{i=1}^{N}$. For the sake of simplicity, we assume a flat-Earth model and we consider line-of-sight Euclidian distances between preferred rendez-vous locations. Even though the actual real-world distance (road, railway, boat, etc.) between two locations is at least as large as their Euclidian distance, the proportion between distances in the real world is assumed to be correlated with the proportion of the respective Euclidian distances. Location priorities, discussed in Section 3.7, can be used for isolated or ranked locations.

We assume that each of the N users has his own public/private key pair $(K_P^{u_i}, K_s^{u_i})$, certified by a trusted CA, which is used to digitally sign/verify the messages that are sent to the LDS. Moreover, we assume that the N users share a common secret that is utilized to generate a shared public/private key pair $(K_P^{M_v}, K_s^{M_v})$ in an online fashion for each meeting setup instance v. The private key $K_s^{M_v}$ generated in this way is known only to all meeting participants, whereas the public key $K_P^{M_v}$ is known to everyone including the LDS. This could be achieved through a secure credential establishment protocol such as in [29, 32, 101].

The LDS executes the FRVP algorithm on the inputs it receives from the users in order to compute the FRV point. The LDS is also able to perform public-key cryptographic functions. For instance, a common public-key infrastructure using the RSA cryptosystem [134] could be employed. Let K_P^{LDS} be the public key, certified by a trusted CA, and K_s^{LDS} the corresponding private key of the LDS. K_P^{LDS} is publicly known and users encrypt their input to the FRVP algorithm using this key; the encrypted input can be decrypted by the LDS using its private key K_s^{LDS}. This ensures message confidentiality and integrity for all the mes-

[1] Throughout this chapter, we use the words *users* and *devices* interchangeably. The meaning is clear from the context, unless stated otherwise.

Table 3.1: Table of symbols.

SYMBOL	DEFINITION	SYMBOL	DEFINITION
$\mathbf{Adv}^{\mathbf{IDT}}$	Identifiability advantage	L_i	Preferred rendez-vous location of user *i*, $L_i = (x_i, y_i)$
$\mathbf{Adv}^{\mathbf{d\text{-}LNK}}$	Distance-linkability advantage	L_{fair}	Fair rendez-vous location
$\mathbf{Adv}^{\mathbf{c\text{-}LNK}}$	Coordinate-linkability advantage	**PS**	Positioning Service
d_{ij}	Euclidian distance between two points L_i, L_j in the plane	u_a	Attacker (a user participating in the PPFRVP protocol)
D_i^M	Maximum Euclidian distance of L_i *i* to any other $L_j, j \neq i$	$E(.)$	Encryption of (.) (the encryption scheme is clear from the context)
f	Public transformation function based on secret key (for privacy)	$ElG(.)$/ $Pai(.)$	Encryption of (.) using the ElGamal/Paillier encryption scheme
g	Optimization function	σ, θ	Element-permutation functions
LDS	Location Determination Server		

sages exchanged between users and the LDS. For simplicity of presentation, in our protocols we do not explicitly show these cryptographic operations involving LDS's public/private key. A complete list of symbols can be found in Table 3.1.

The PS (or Positioning Services) is a service that allows users to determine their preferred (or own) position by using a third-party localization service, if they are not able to determine such positions locally on the devices. The PS is also required to determine the geographic coordinates of a given POI within a region (e.g. a known restaurant). In order to limit the information that the PS learns about the users' location requests, a private information retrieval technique, such as [65], can be used. This would effectively prevent the PS from knowing which POIs the users have requested and thus protect their location privacy *vis-à-vis* the PS.

Adversarial Model

Location Determination Server The LDS is assumed to execute the algorithms correctly, i.e., take all the inputs and produce the output according to the algorithm. However, the LDS may try to learn information about users' location preferences from the received inputs, the intermediate results and the produced outputs. This type of adversarial behavior is usually referred to as *honest-but-curious* adversary (or semi-honest) [68]. In most practical settings, where service providers have a commercial interest in providing a faithful service to their customers, the assumption of a semi-honest LDS is generally sufficient.

Users The participating users also want to learn the private location preferences of other users from the output of the algorithm they receive from the LDS. We refer to such attacks as passive attacks. As user inputs are encrypted with the LDS's public key K_P^{LDS}, there is a confidentiality guarantee against basic eavesdropping by participants and non participants. In addition to these attacks, participants may also attempt to actively attack the protocol by colluding with

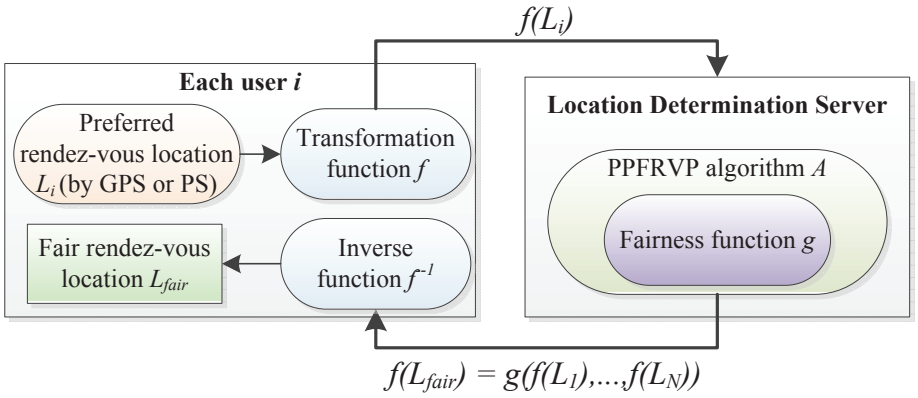


Figure 3.1: Functional diagram of the PPFRVP protocol, where the PPFRVP algorithm is executed by an LDS.

other users or manipulating their own inputs to learn the output.

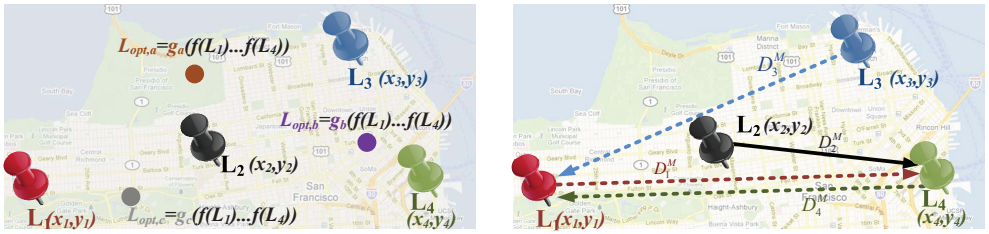
3.3 Problem Definition

In this work, we consider the problem of finding, in a privacy-preserving way, the rendez-vous point among a set of user-proposed locations, such that (i) the rendez-vous point is a point that is *fair* (as defined in Section 3.4.1) with respect to the given locations, (ii) each of the users gets to know only the final rendez-vous location and (iii) no participating user or third-party server learns private location information about any other user involved in the computations. We refer to an algorithm that solves this problem as *Privacy-Preserving Fair Rendez-Vous Point (PPFRVP)* algorithm. In general, any PPFRVP algorithm A should accept the inputs and produce the outputs, as described below.

- *Input*: a transformation f of private locations L_i: $f(L_1)||f(L_2)||\ldots||f(L_N)$. where f is a one-way public function (based on secret key) such that is hard (success with only a negligible probability) to determine the input L_i without knowing the secret key, by just observing $f(L_i)$.

- *Output*: an output $f(L_{fair}) = g(f(L_1), \ldots, f(L_N))$, where g is a fairness function and $L_{fair} = (x_l, y_l) \in \mathbb{N}^2$ is the fair rendez-vous location that has been selected for this particular set of users, such that it is hard for the LDS to determine L_{fair} by just observing $f(L_{fair})$. Given $f(L_{fair})$, each user should be able to compute $L_{fair} = f^{-1}(f(L_{fair}))$ using his local data.

Figure 3.1 shows a functional diagram of a PPFRVP protocol, where the PPFRVP algorithm A is executed by an LDS. The fairness function g can be defined in several ways, depending on the preferences of users or policies. For



(a) General PPFRVP scenario, where distinct fairness functions g_a, g_b, g_c output three different fair rendez-vous locations $L_{opt,a}, L_{opt,b}, L_{opt,c}$, given the user-preferred rendez-vous locations $L_1, \ldots, L_4$.

(b) PPFRVP scenario, where the fairness function is $g = \text{argmin}_i(D_i^M)$. The dashed arrows represent the maximum distance D_i^M from each user u_i to any user $j \neq i$, whereas the solid line is the minimum of all such maximum distances. The fair rendez-vous location is $L_{fair} = L_2 = (x_2, y_2)$.

Figure 3.2: Possible PPFRVP problem solutions for the general g fairness function 3.2(a) and a specific, minimum distance-based g function 3.2(b).

instance, users might prefer to meet in locations that are close to their offices, and their employers might prefer a place that is closest to their clients. Figure 3.2(a) shows three different optimal locations for three distinct fairness functions g.

In Section 3.4.1 we describe one such fairness function that minimizes the maximum displacement of any user to all other locations. Such function is globally fair and general enough, and it can be extended to include more constraints and parameters.

3.4 Privacy-Preserving Location Determination

In the previous section, we have defined the fundamental building blocks that constitute a PPFRVP problem, both from a functional perspective (as in Figure 3.1) and from a privacy context. From a practical point of view, however, the problem is to design specific solutions and protocols that can be implemented on existing commercial mobile devices. In order to achieve the integration between resource-constrained mobile devices and the existing client-server network paradigm, our solutions have to be efficient in terms of computations and communication complexities, while taking advantage of the increasingly available communication bandwidth on mobile devices.

In this section, we present our solution to the PPFRVP problem. First, we discuss the mathematical tools that we use in order to model the fairness function g and the transformation functions f.In the following subsections, we define the fairness function g by taking advantage of the properties of three well-known cryptographic primitives that are used to implement the transformation function

f. These primitives, in turn, will guarantee that no private information about the preferred locations of any user is leaked to any other user or third-party involved in the computations. Finally, by merging the f and g components of the PPFRVP algorithm, we design our complete PPFRVP protocol. We then analytically evaluate its privacy properties and its computation and communication complexities.

In order to separate the optimization part of the PPFRVP algorithm A from its implementation using cryptographic primitives, we first discuss the fairness function g and then the transformation function f.

3.4.1 Fairness Function g

In order to determine a *fair* rendez-vous location, there are several factors that need to be considered. First, the optimality criterion needs to consider the spatial constraints present in the problem. For example, a rendez-vous location $L_{fair} = (x_l, y_l)$ among N users $\mathbb{U} = \{u_i\}_{i=1}^{N}$ might be fair when all users can reach L_{fair} in a "reasonable" amount of time. Another criterion might be to minimize the total displacement of all users in order to reach L_{fair}, or simply making sure that no user is "too far" from L_{fair} with respect to another user. Second, computing an fair rendez-vous location in a privacy-preserving way requires g to perform optimization operations in an oblivious fashion. In this work, we achieve this by using the properties of appropriate cryptographic schemes. Features such as homomorphic encryption and semantic security are of particular interest, as they allow operations on the plain text elements to be accomplished by oblivious computations on encrypted elements.

In this work, we consider the fairness criterion that has been widely used in operations research to solve the *k-center* problem. In the k-center problem, the goal is to find $L_1, \ldots, L_k$ locations among N given possible places, in order to optimally place k facilities, such that the maximum distance from any place to its closest facility is minimized. For a two dimensional coordinate system, the Euclidian distance metric is usually employed.

As the PPFRVP problem consists in determining the fair rendez-vous location from a set of user-preferred locations, we focus on the *k-center* formulation of the problem with $k = 1$. This choice is also grounded on the fact that not choosing L_{fair} from one of the location preferences $L_1, \ldots, L_N$ might potentially result in a location L_{fair} that is not suited for the kind of meeting that the participants require. The solution can easily be extended or integrated with mapping applications (on the users' devices) so that POIs around L_{fair} are automatically suggested for the meeting. Figure 3.2(b) shows an example PPFRVP scenario modeled as a k-center problem, where four users want to determine the fair rendez-vous location L_{fair}.

The k-center formulation considers the Euclidian distances, but it does not encompass other fairness parameters, such as accessibility of a place and the means of transportation. In this work, we focus on the pure k-center formulation as the essential building block of a more complete model, which we extend to encompass multiple and prioritized preferences in Section 3.7.

Let $d_{ij} \geq 0$ be the Euclidian distance between two points $L_i, L_j \in \mathbb{N}^2$, and $D_i^M = \max_{j \neq i} d_{ij}$ be the maximum distance from L_i to any other point L_j. Then, the PPFRVP problem can be formally defined as follows.

Definition 3.1. *The* PPFRVP *problem is to determine a location* $L_{fair} \in \mathbb{L} = \{L_1, \ldots, L_N\}$, *where* $fair = arg\min_i D_i^M$

A solution for the PPFRVP problem finds, in a privacy-preserving way, the fair rendez-vous location among the set of proposed (and user-desired) locations, such that the distance from the furthest proposed location to the fair one is minimized.

There are two important steps involved in the computation of the fair location L_{fair}. The first step is to compute the pairwise distances d_{ij} among all preferred locations $L_i, L_j \in \{L_1, \ldots, L_N\}$. The second step requires the computations of the maximum and minimum values of such distances. Before proceeding with these computations, in the following subsection we examine the features provided by the cryptographic functions that will ensure the privacy of individual user-desired locations L_i, $\forall i = 1, \ldots, N$.

3.4.2 Transformation Functions f

The fairness function g requires the computation of two functions on the private user-preferred locations L_i: (i) the distance between any two locations $L_i \neq L_j$ and (ii) the minimum of the maximum of these distances. In order to solve the FRVP problem and to preserve privacy, we rely on computationally secure cryptographic primitives. In our protocol, we consider three such schemes: the *Boneh-Goh-Nissim* (BGN) [24], the *ElGamal* [49] and the *Paillier* [124] encryption schemes.

There are several cryptographic schemes that can be used, but not all of them provide the same features. We are interested in using secure schemes that allow us to compute the Euclidian distance between two points in the plane and the maximization/minimization functions. What makes these schemes useful in our setting are their homomorphic encryption properties. Given two plain texts m_1, m_2 with their respective encryptions $E(m_1), E(m_2)$, the multiplicative property (possessed by the ElGamal and partially by the BGN schemes) states that $E(m_1) \odot E(m_2) = E(m_1 \cdot m_2)$, where $\odot$ is an arithmetic operation in the encrypted domain that is equivalent to the usual multiplication operation in the plain text domain. The additive homomorphic property (possessed by the BGN

and the Paillier schemes) states that $E(m_1) \oplus E(m_2) = E(m_1 + m_2)$, where $\oplus$ is an arithmetic operation in the encrypted domain which is equivalent to the usual sum operation in the plain text domain. Details about the initialization, operation and security of the encryption schemes can be found in [49, 24, 124]. Hereafter we describe the main operational aspects the BGN scheme; in Section 2.4 we already described the properties for the ElGamal and Paillier encryption schemes.

BGN Given two plaintexts $m_1, m_2 \in \mathbb{Z}_T^*$ (where $T < q$ and q is a large prime) with their respective encryptions $E(m_1), E(m_2)$, the BGN possesses the following *multiplicative* and *additive* homomorphic properties

$$E(m_1 \cdot m_2) = e(E(m_1), E(m_2)) \cdot h_1^r \mod n$$
$$E(m_1 + m_2) = E(m_1) \cdot E(m_2) \cdot h^r \mod n$$

where $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ is an admissible bilinear map, $\mathbb{G}, \mathbb{G}_1$ are two bilinear groups of composite order $n = pq$ (p, q are two large primes), h, g are public, $h_1 = e(g, h)$ and $r \in \mathbb{Z}_n$ is a random integer. BGN is an elliptic curve-based scheme and therefore much shorter keys can be used compared to ElGamal and RSA. A 160-bit key in elliptic curve cryptosystems is generally believed to provide equivalent security as a 1024-bit key in RSA and ElGamal [135]. However, due to the construction of the BGN scheme, only one homomorphic multiplication on each encrypted element is allowed, whereas an infinite number of homomorphic additions can be performed.

Based on the three aforementioned encryption schemes, we now describe the distance computation algorithms that are used in our solution.

3.4.3 Distance Computations

To determine the fair rendez-vous point, we need to find the location L_{fair}, where $fair \in \{1, \ldots, N\}$, that minimizes the maximum distance between any user-preferred location and L_{fair}. In our algorithms, we work with the *square* of the distances, as they are much easier to compute in an oblivious fashion using the homomorphic properties of the encryption schemes. The problem of finding the argument that minimizes the maximum distance is equivalent to finding the argument that minimizes the maximum *squared* distance (provided that all distances are greater than 0). As the squaring function maintains the relative order, the algorithm is still correct.

Hereafter we propose two distance computation modules that will be used in our PPFRVP protocol. Each of these modules computes the square of all pairwise distances between any two user-desired locations, and preserves the privacy of each user's preferred location L_i, $\forall i \in \{1, \ldots, N\}$.

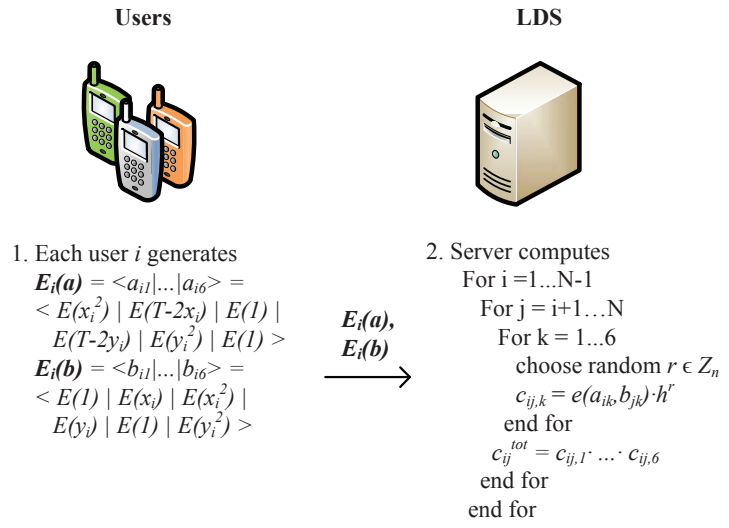


Figure 3.3: Distance computation protocol based on the BGN encryption scheme.

BGN-distance

Our first distance computation algorithm is based on the BGN encryption scheme. and is shown in Figure 3.3. This novel protocol requires only one round of communication between each user and the LDS, and it efficiently uses both the multiplicative and additive homomorphic properties of the BGN scheme. The BGN-distance protocol works as follows. In Step 1, each user u_i, $\forall i \in \{1, \ldots, N\}$, creates the vectors

$$E_i(a) =< a_{i1}| \ldots |a_{i6} >=< E(x_i^2)|E(T-2x_i)|E(1)|E(T-2y_i)|E(y_i^2)|E(1) >$$
$$E_i(b) =< b_{i1}| \ldots |b_{i6} >=< E(1)|E(x_i)|E(x_i^2)|E(y_i)|E(1)|E(y_i^2) >$$

where $E(.)$ is the encryption of $(.)$ using the BGN scheme with the fresh session key $K_P^{M_v}$ and $L_i = (x_i, y_i)$ is the desired rendez-vous location of user u_i. Afterwards, each user sends the two vectors $E_i(a), E_i(b)$ over a secure channel to the LDS. In Step 2, the LDS computes the scalar product of the received vectors by first applying the multiplicative and then the additive homomorphic property of the BGN scheme. For example, in a scenario with two users, one can easily verify that

$$\begin{aligned} E_i(a) \bullet E_j(b) &= E(x_i^2 + x_j(T-2x_i) + x_j^2 + y_j(T-2y_i) + y_i^2 + y_j^2 \mod T) \\ &= E(d_{ij}^2 \mod T) \end{aligned}$$

where T is chosen such that $\forall i, j \in \{1, \ldots, N\}, d_{ij}^2 < T$. At this point, the LDS has obliviously computed $E(d_{ij}^2)$, which is the (encrypted) square of the pairwise distances between all pairs L_i, L_j of user-desired locations, where $i \neq j$.

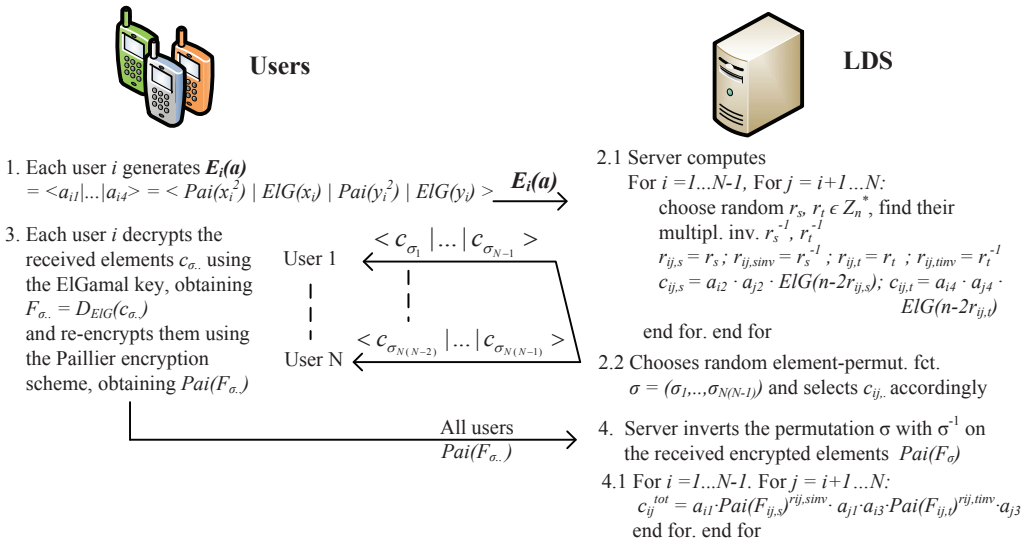


Figure 3.4: Distance computation protocol based on the ElGamal and Paillier encryption schemes.

Paillier-ElGamal-distance

An alternative scheme for the distance computation is based on both the Paillier and ElGamal encryptions, as shown in Figure 3.4. As neither Paillier or ElGamal possess both multiplicative and additive properties, the resulting algorithm requires one extra step in order to achieve the same result as the BGN-based scheme, i.e., obliviously computing the pairwise squared distances d_{ij}^2. The distances are computed as follows. In Step 1, each user u_i, $\forall i \in \{1, \ldots, N\}$, creates the vector

$$E_i(a) =< a_{i1} | \ldots | a_{i4} >=< Pai(x_i^2) | ElG(x_i) | Pai(y_i^2) | ElG(y_i) >$$

where $Pai(.)$ and $ElG(.)$ refer to the encryption of $(.)$ using the Paillier or ElGamal encryption schemes, respectively. Afterwards, each user u_i sends the vector $E_i(a)$ to the LDS, encrypted with LDS's public key. In step 2.1, the LDS computes the scalar product of the second and fourth element of the received vectors (as shown in Figure 3.4). In order to hide this intermediate result from the users, the LDS obliviously randomizes these results with random values r_s, r_t. At the same time, the LDS computes the multiplicative inverse of such values, denoted as r_s^{-1} and r_t^{-1} respectively. These randomized scalar products are denoted as $c_{ij,s}$ and $c_{ij,t}$. In step 2.2, the LDS permutes the order of all $c_{ij,s}$ and $c_{ij,t}$ with its private element-permutation function $\sigma = [\sigma_1, \ldots, \sigma_{N(N-1)}]$, and sends N such distinct elements to each user u_i. In step 3, each user simply decrypts the received elements with the ElGamal private key and re-encrypts them with the Paillier public key. Then each user sends the re-encrypted elements to the LDS in the

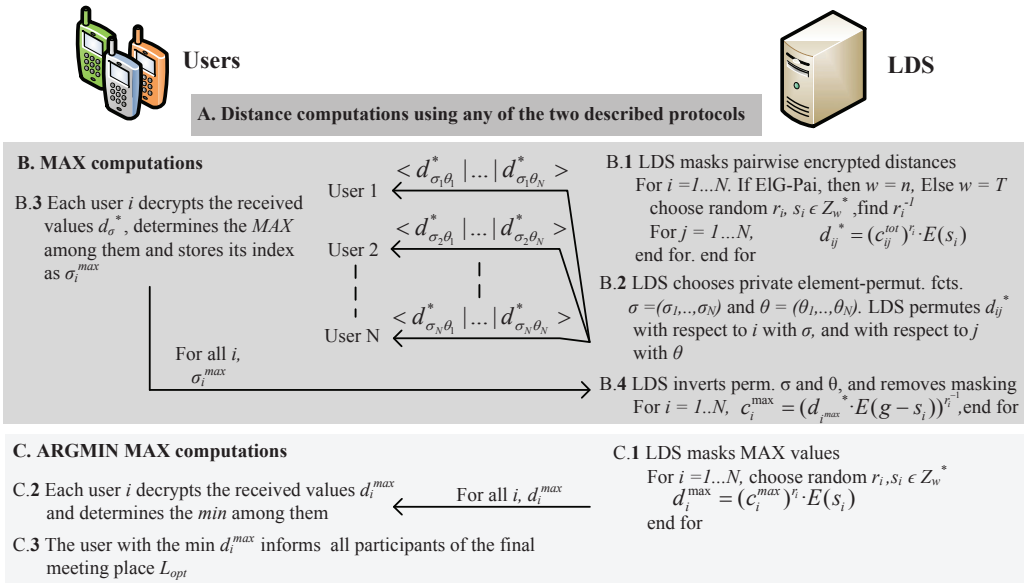


Figure 3.5: Privacy-Preserving Fair Rendez-Vous Point (PPFRVP) protocol.

same order as he received it. In step 4, the LDS reverts the element-permutation function σ, and in step 4.1 it finally computes the d_{ij}^2 for all i, j, after having removed the randomizing factors $r_{ij,s}, r_{ij,t}$ with their inverses $r_{ij,sinv}$ and $r_{ij,tinv}$. At this point, the LDS has securely computed $E(d_{ij}^2)$, the (encrypted) square of the pairwise distances between all pairs of user-desired locations $L_i \neq L_j$.

As the ElGamal-Paillier based distance computation involves decryption/re-encryption operations, it may be possible for participants to maliciously change the masked values. For instance, such an active attack could be performed in order to disrupt the distance computations or to manipulate the result for personal advantage (such as a personally convenient but generally subfair rendez-vous location). We discuss such active attacks in Section 3.4.6.

3.4.4 The PPFRVP Protocol

In the previous subsections, we defined all the necessary operations and cryptographic tools that are required in order to solve the PPFRVP problem. We now describe our protocol for the PPFRVP problem, as shown in Figure 3.5. The protocol has three main modules: (A) the distance computation module, (B) the MAX module and (C) the ARGMIN MAX module.

Distance computations

The first module (distance computation) uses one of the two protocols defined in the previous subsection (BGN-distance or Paillier-ElGamal-distance). We note that modules (B) and (C) use the same encryption scheme as the one used in

module (A). In other words, $E(.)$ of Figure 3.5 refers to the encryption of (.) using either the BGN or the Paillier encryption scheme. Once the distance protocol has been decided, the next modules (B) and (C) are executed as follows.

MAX computations

In Step B.1, the LDS needs to obliviously hide the values within the encrypted elements (i.e., the pairwise distances computed earlier), before sending them to the users, in order to avoid leaking any kind of private information such as the pairwise distance or desired locations to any user.[2] In order to obliviously mask such values, for each index i the LDS generates two random values r_i, s_i that are used to scale and shift the c_{ij}^{tot} (the encrypted square distance between L_i, L_j) for all j, obtaining d_{ij}^*. This is done in order to (i) ensure privacy of real pairwise distances, (ii) be resilient in case of collusion among users and (iii) preserve the internal order (the inequalities) among the pairwise distance from each user to all other users. Afterwards, in Step B.2 the LDS chooses two private element-permutation functions σ (for i) and θ (for j) and permutes d_{ij}^*, obtaining the permuted values $d_{\sigma_i\theta_j}^*$, where $i, j \in \{1, \ldots, N\}$. The LDS sends N such distinct elements to each user. In Step B.3, each user decrypts the received values, determines their maximum and sends the index σ_i^{max} of the maximum value to the LDS. In Step B.4 of the MAX module (B), the LDS inverts the permutation functions σ, θ and removes the masking from the received indexes corresponding to the maximum distance values.

ARGMIN MAX computations

In Step C.1, the LDS masks the true maximum distances by scaling and shifting them by the same random amount, such that their order (the inequalities among them) is preserved. Then the LDS sends to each user all the masked maximum distances. In Step C.2 each user decrypts the received masked (randomly scaled and shifted) maximum values, and determines the minimum among all maxima. In Step C.3, each user knows which identifier corresponds to himself, and the user whose preferred location has the minimum distance sends to all other users the fair rendez-vous location in an anonymous way.

After the last step, each user receives the final fair rendez-vous location, but no other information regarding non-fair locations or distances is leaked. In the next section, we analyze our PPFRVP protocols with respect to their privacy and

[2] After the distance computation module (A), the LDS possesses all encrypted pairwise distances. This encryption is made with the public key of the participants and thus the LDS cannot decrypt the distances without the corresponding private key. The oblivious (and order-preserving) masking performed by the LDS at Step B.1 is used in order to hide the pairwise distances from the users themselves, as otherwise they would be able to obtain these distances and violate the privacy of the users.

complexity aspects. In order to assess the efficiency and to know whether the proposed PPFRVP protocol fulfills the privacy requirements defined in Section 3.3, we present the complexity and privacy analysis in the next subsection.

3.4.5 Privacy Requirements and Definitions

Informally, the privacy requirements can be stated as follows. After the execution of the PPFRVP algorithm, any user u_i should not be able to infer (i) the preferred location L_j of any other user $u_j \neq u_i$ nor (ii) the relative distances d_{ij} between any two users $u_i \neq u_j$. Likewise, any LDS (and PS) should not be able to infer (iii) the preferred location L_i of any user u_i, (iv) the relative distance d_{ij} between any two users $u_i \neq u_j$ nor (v) the final rendez-vous location L_{fair}. Such privacy requirements can be grouped in two components, called as *user-privacy* and *server-privacy*, formally defined as follows.

User-Privacy

The *user-privacy* of any PPFRVP algorithm A measures the probabilistic advantage that an attacker a (a user participating in the PPFRVP protocol or an external user) gains towards learning the preferred location L_j of at least one other user $j \in \{1, \ldots, N\}$, except the final fair rendez-vous location L_{fair}, after all users have participated in the execution of the PPFRVP protocol. Clearly, an external user does not learn about any preferred locations as it does not receive the output of the algorithm. Therefore, we only consider the non-trivial case of users participating in the PPFRVP protocol as attackers, i.e., u_a where $a \in \{1, \ldots, N\}$.

We express the user-privacy in terms of three adversary advantages. First, we measure the *identifiability advantage*, which is the probabilistic advantage of u_a in correctly guessing the preferred location L_i of any user $u_i \neq u_a$. We denote it as $Adv_a^{IDT}(A)$. Second, we measure the *distance-linkability advantage*, which is the probabilistic advantage of u_a in correctly guessing whether the distance d_{ij} between any two users $u_i \neq u_j$, is greater than a given parameter s, without necessarily knowing any users' preferred locations L_i, L_j. We denote this advantage as Adv_a^{d-LNK}. Finally, we measure the *coordinate-linkability advantage*, which is the probabilistic advantage of u_a in correctly guessing whether a given coordinate x_i (or y_i) of a user u_i is greater than the corresponding coordinate(s) of another user $u_j \neq u_i$, i.e., x_j (or y_j), without necessarily knowing any users' preferred locations L_i, L_j. We denote this advantage as Adv_a^{c-LNK}. The next observation follows from the above definitions.

Observation 3. *If an adversary has an identifiability advantage over any two distinct users $u_i \neq u_j$, this implies it has distance- and coordinate-linkability*

advantages over those two users as well. However, the inverse is not necessarily true.

We semantically define the identifiability and linkability advantages by using a challenge-response methodology, which has been widely used to prove the security of cryptographic protocols. We now describe such a challenge-response game for the identifiability advantage $Adv_a^{IDT}(A)$ of any adversary u_a in a PPFRVP algorithm A. Let $\mathbb{U} = \{u_1, \ldots, u_N, u_a\}$ be the set of all users, including the adversary u_a, and let $\mathbb{C}$ be the challenger. The game is defined as follows:

1. Challenger setup: $\mathbb{C}$ privately collects the preferred rendez-vous locations L_i; $L_i \neq L_j, \forall i, j \in \{1, \ldots, N\}$.
2. Algorithm execution: $\mathbb{C}$ executes the PPFRVP algorithm A with all users $\mathbb{U}$ and computes $f(L_{fair}) = g(f(L_1), \ldots, f(L_N), f(L_a))$. It then sends $f(L_{fair})$ to each user $u_i \in \mathbb{U}$.
3. Challenge: $\mathbb{C}$ chooses a random $k \in \{1, \ldots, N\}$ and sends L_k to the adversary u_a.
4. Guess: u_a chooses a value $k' \in \{1, \ldots, N\}$ and sends it back to the challenger. u_a wins the game if $k' = k$, otherwise he loses.

The identifiability advantage is defined as the probabilistic advantage of the adversary in winning this game. It is denoted as:

$$Adv_a^{IDT}(A) = Pr[u_a \text{ wins the game}] - 1/N = Pr[k' = k] - 1/N \tag{3.1}$$

where $Pr(k' = k)$ is the probability that u_a correctly guesses the value k chosen by the challenger.

The above notion of identifiability is also called *weak identifiability* because the adversary knows that the challenge belongs to one of the participant. A stronger notion of identifiability can also be defined. In the definition of *strong identifiability*, the challenge (in Step 3) is a randomly chosen non-trivial two dimensional position coordinate, instead of restricting the challenge location to belong to one of the participating user. The adversary in this game wins if he correctly guesses if the challenge location belongs to one of the participants or not. In this work, we focus only on the weak identifiability property.

Similarly, we define the distance-linkability advantage $Adv_a^{d-LNK}(A)$ of any adversary u_a in a PPFRVP algorithm A by means of the following game.

1. Challenger setup: $\mathbb{C}$ privately collects the preferred rendez-vous locations $L_i \neq L_j, \forall i, j \in \{1, \ldots, N\}$.
2. Algorithm execution: $\mathbb{C}$ executes the PPFRVP algorithm A with all users $\mathbb{U}$ and computes $f(L_{fair}) = g(f(L_1), \ldots, f(L_N), f(L_a))$. It then sends $f(L_{fair})$ to each user $u_i \in \mathbb{U}$.

3. Challenge: $\mathbb{C}$ chooses a random value s and two distinct users $u_j, u_k, \forall j, k \in \{1, \ldots, N\}, j \neq k$. $\mathbb{C}$ sends (j, k, s) to the adversary.

4. Guess: u_a responds with a value $s^* \in \{0, 1\}$. u_a wins the game if $s^* = 0$ and $d_{j,k} \geq s$, or if $s^* = 1$ and $d_{j,k} < s$. Otherwise, the adversary looses.

The distance-linkability advantage is defined as the probabilistic advantage of the adversary in winning this game. It is denoted as:

$$Adv_a^{d-LNK}(A) = Pr[s^* = 0 \wedge d_{j,k} \geq s] + Pr[s^* = 1 \wedge d_{j,k} < s] - 1/2 \qquad (3.2)$$

Finally, we define the coordinate-linkability advantage $Adv_a^{c-LNK}(A)$ of any adversary u_a in a PPFRVP algorithm A by means of the following game.

1. Challenger setup: $\mathbb{C}$ privately collects the preferred rendez-vous locations $L_i \neq L_j, \forall i, j \in \{1, \ldots, N\}$.

2. Algorithm execution: $\mathbb{C}$ executes the PPFRVP algorithm A with all users $\mathbb{U}$ and computes $f(L_{fair}) = g(f(L_1), \ldots, f(L_N), f(L_a))$. It then sends $f(L_{fair})$ to each user $u_i \in \mathbb{U}$.

3. Challenge: $\mathbb{C}$ throws an unbiased coin to select a coordinate axis $b \in \{x, y\}$. $\mathbb{C}$ randomly chooses $j, k \in \{1, 2, \ldots, N\}, j \neq k$. $\mathbb{C}$ sends $\{j, k, b\}$ to u_a as a challenge.

4. Guess: u_a responds with a value $r \in \{0, 1\}$ and sends it back to the challenger. u_a wins the game if:

$$\begin{cases} r = 0 \text{ and } b_j \leq b_k \text{ OR} \\ r = 1 \text{ and } b_j > b_k \end{cases}$$

u_a looses the game otherwise. T

The coordinate-linkability advantage is defined as the probabilistic advantage of the adversary in winning this game. It is denoted as:

$$Adv_a^{c-LNK}(A) = Pr[r = 0 \wedge b_j \leq b_k] + Pr[r = 1 \wedge b_j > b_k] - 1/2 \qquad (3.3)$$

For the third-party (LDS) adversary, the game definitions are similar to those of the user adversary, except that the LDS does not receive L_{fair} at the Step 2 of the game. As the LDS obtains no additional information compared to u_a but even less, the probability that it guesses correctly in Step 4 cannot be higher than that of u_a and not lower than a random guess. Thus, the LDS cannot win the game with non-negligible advantage.

We now define the user-privacy of any PPFRVP algorithm A on a per-execution basis in the following way.

Definition 3.2. *An execution of the PPFRVP algorithm A is* user-private *if the identifiability advantage* $Adv_a^{IDT}(A)$*, the distance-linkability advantage* $Adv_a^{c-LNK}(A)$ *and the coordinate-linkability advantage* $Adv_a^{c-LNK}(A)$ *of each participating user* $u_i, i \in \{1, \ldots, N\}$ *are negligible in terms of the* number *of user-preferred rendez-vous locations* L_i*.*

In general, a function $f(x)$ is called negligible if, for any positive polynomial $p(x)$, there is an integer B such that for any integer $x > B, |f(x)| < 1/p(x)$ [68]. According to Definition 3.2, an execution of the PPFRVP algorithm is user-private if and only if any user u_a does not gain any (actually, negligible) additional knowledge about the preferred rendez-vous locations L_j of any other user $u_j \neq u_a$, except the value of the final fair rendez-vous location L_{fair}.

Server-Privacy

The *server-privacy* of any PPFRVP algorithm A measures the probabilistic advantage that the LDS gains in learning the preferred rendez-vous locations L_i of any user u_i, $i \in \{1, \ldots, N\}$. As in the case of user-privacy, we express the server-privacy by means of three advantages. First, we measure the probabilistic advantage of an LDS in correctly guessing the preferred location L_i of any user u_i, called *identifiability advantage* and denoted as $Adv_{LDS}^{IDT}(A)$. Second, we measure the probabilistic advantage of an LDS in correctly guessing whether the distance d_{ij} between any two users $u_i \neq u_j$ is greater than a given parameter s, without necessarily knowing any users' preferred locations L_i, L_j. We call this the *distance-linkability advantage* and we denote it as $Adv_{LDS}^{d-LNK}(A)$. Third, we measure the probabilistic advantage in correctly guessing whether a given coordinate x_i (or y_i) is greater than the same coordinate of another user $j \neq i$, i.e., x_j (or y_j), without necessarily knowing any users' preferred locations L_i, L_j. We call this the *coordinate-linkability advantage* and we denote it as $Adv_{LDS}^{c-LNK}(A)$.

Definition 3.3. *An execution of the PPFRVP algorithm A is* server-private *if the identifiability advantage* $Adv_{LDS}^{IDT}(A)$*, the distance-linkability advantage* Adv_{LDS}^{c-LNK} *and the coordinate-linkability advantage* Adv_{LDS}^{c-LNK} *of an LDS are negligible.*

However, it is reasonable to assume that in practice users will be able to perform multiple executions of the PPFRVP protocol, possibly with different sets of participating users at each time. This is particularly true if such a meeting-location service is offered, for instance, by providers to their subscribers. As a consequence, privacy of a PPFRVP should be defined over multiple executions.

PPFRVP Privacy

We now formally express the privacy conditions that any PPFRVP algorithm A has to satisfy, based on the above definitions. First, we define a *private* execution

of a PPFRVP algorithm as follows.

Definition 3.4. *A* private *execution of any PPFRVP algorithm A is an execution which does not reveal more information than what can be derived from the inputs, the intermediate results and its output.*

Based on how memory is retained over sequential executions, we define two types of algorithm executions, namely, dependent and independent.

Definition 3.5. *An* independent *(respectively* dependent*)* execution *is a single private execution of the PPFRVP algorithm defined in Section 3.3 in which* no *(respectively* some*) information of an earlier and current execution is retained and passed to future executions.*

The information that might be transferred from an earlier execution to the next can include past inputs to the algorithm, intermediate results (on the LDS) and the outputs of the algorithm. Based on the type of execution, the privacy conditions of a privacy-preserving meeting-location algorithm can be defined as follows.

Definition 3.6. *A PPFRVP algorithm A is* execution *(respectively* fully*) privacy-preserving if and only if for every* independent *(respectively* all*) execution(s)*

1. *A is correct; All users are* correctly *able to compute the final fair rendezvous location* L_{fair}*;*
2. *A is* user-private*;*
3. *A is* server-private.

A fully privacy-preserving meeting-location (PPFRVP) algorithm is a much stronger (and difficult to achieve) privacy requirement. In this work, we focus on achieving execution privacy. The relationship between a fully PPFRVP and execution PPFRVP algorithm is given by the following observation.

Observation 4. *Any PPFRVP algorithm A, as defined in Section 3.3, is execution privacy-preserving if it is fully privacy-preserving, but the inverse is not true.*

3.4.6 Analytical Evaluation

The proposed PPFRVP protocol, shown in Figure 3.5, is based on the interaction between users and a third-party LDS. Each of these parties performs operations on both plaintext and encrypted elements, and the resources available on the user devices are usually lower than those of the LDS. We now present the privacy and complexity analysis of our protocols, and then we show the asymptotic complexities of our two distance and PPFRVP protocols, by considering both client and LDS computation, communication and memory complexities.

Privacy Analysis

In our analysis, we consider two types of adversaries: Passive (honest-but-curious) and active adversaries. The passive try to learn as much information as possible from their inputs, the execution of the PPFRVP protocol and its output, without maliciously injecting or modifying data. The active adversaries, on the contrary, try on purpose to manipulate the data in order to obtain private information.

The aforementioned definition captures the privacy requirements of a single execution of a PPFRVP algorithm. By repeated interactions among a stable set of users, L_{fair} could be used to infer possible L_i of other users. The issue of *learning* from repeated interaction is inherent to any algorithm that, based on a set of private inputs, chooses one of them in particular, based on some criterion. For this reason, in this work we consider privacy for a single execution of the PPFRVP algorithm, or for repeated executions but with different sets of users.

Passive Adversary Under the passive adversary model, we have the following result.

Proposition 3.1. *The proposed PPFRVP protocols are correct and they guarantee identifiability- and coordinate-linkability privacy. However, they do not guarantee distance-linkability privacy.*

In other words, Proposition 3.1 states that both the proposed algorithms correctly compute the fair rendez-vous location, given the received inputs, and that they do not reveal any users' preferred rendez-vous locations to any other user, except the fair rendez-vous location L_{fair}. However, in the following proof we show that there is at least one case where the adversary can win the distance-linkability game with non-negligible advantage. Nevertheless, both algorithms do not reveal information about the location or relationship between the coordinates of any $L_i, \forall i \in \{1, \ldots, N\}$. The LDS does not learn any information at all about any user-preferred locations.

Proof. **Correctness**
Given the encrypted set of user-preferred locations $f(L_1), \ldots, f(L_N)$, the proposed PPFRVP algorithms compute the pairwise distance between each pair of users d_{ij}, $\forall i, j \in \{1, \ldots, N\}$, according to the schemes of the respective distance computation algorithms. Following the sequence of steps for such computation, one can easily verify that the ElGamal-Paillier based distance computation algorithm computes

$$\begin{aligned} Pai(d_{ij}^2) = Pai(x_i^2) \cdot Pai(-2x_i x_j) \cdot Pai(y_j^2) \cdot Pai(y_i^2) \cdot Pai(-2y_i y_j) \\ \cdot Pai(y_j^2) = Pai(x_i^2 - 2x_i x_j + x_j^2 + y_i^2 - 2y_i y_j + y_j^2) \end{aligned}$$

which is the same result that is achieved by the BGN-based distance algorithm.

After the pairwise distance computations, the PPFRVP algorithm computes the masking of these pairwise distances by scaling and shifting operations. The scaling operation is achieved by exponentiating the encrypted element to the power of r_i, where $r_i \in \mathbb{Z}_w^*$ is a random integer and r_i^{-1} is its multiplicative inverse. The shifting operation is done by multiplying the encrypted element with the encryption (using the public key of the users) of another random integer s_i privately chosen by the LDS. These two algebraic operations mask the values d_{ij}^2 (within the encrypted elements), such that the true d_{ij}^2 are hidden from the users. Nevertheless, thanks to the homomorphic properties of the encryption schemes, the LDS is still able to remove the masking (after the users have identified the maximum value) and correctly re-mask all maxima, such that each user is able to correctly find the minimum of all maxima.

In the end, each user is able to determine L_{fair}, where $fair = \text{argmin}_i \max_j d_{ij}^2$ from the outputs of the PPFRVP algorithm, and therefore the PPFRVP algorithms are correct.

User Identifiability Advantage Hereafter we provide sketches of the proofs of user-privacy, after a private execution of the PPFRVP algorithm A. A sketch is usually given to intuitively show how the formal proof can be constructed with the argument presented in the sketch. In particular, the following sketches are exhaustive, i.e., they cover all possible cases, and they are used to show whether the different advantages are non-negligible and thus whether a PPFRVP algorithm A is execution privacy-preserving.

In the identifiability advantage, there are only two possible outcomes of the PPFRVP algorithm, depending on users' preferred locations L_i: The first case is when $L_{fair} = L_a$, i.e., when the fair rendez-vous location is the one proposed by the adversary; the second case is when $L_{fair} \neq L_a$, i.e., when the fair location is different from the one proposed by the adversary. Hereafter we split the sketch of our proof according to these two (and only possible) cases, and show that the advantage of the adversary is negligible in both these cases:

1. $\mathbf{L_{fair} = L_a}$: In this case, the adversary does not learn any additional information that was not already known to him before the execution of the protocol, except the order among the maximum distances between the users and the corresponding indices. Moreover, we consider here the non-trivial case where the challenger chooses a value $k \neq a$, otherwise the correct answer to the challenge is trivial. It should be noted that the challenger cannot select the trivial case with a probability greater than 1/N (during the challenge step or step 3). In this non-trivial case, the adversary cannot guess the value $k \neq a$ with a higher certainty than he would by a random guess because only the LDS knows the secret scaling and shifting values used for the masking operation. In fact, the order among the masked distances does not reveal any additional information about the actual locations, as there

could be infinitely many locations at the same masked distance. Thus, the advantage of the adversary in this case is negligible.

2. $\mathbf{L_{fair} \neq L_a}$: In this case, the adversary learns, after the execution of the protocol, another preferred location $L_{fair} \neq L_a$ different from his own, in addition to the order among the maximum distances for all users. The adversary is able to compute the distance $d_{a,fair}$ between his preferred location and L_{fair}. However, thanks to the masking operation on the distances and to the independence among the users' preferred locations, the adversary has no additional knowledge to link $d_{a,fair}$ to any other masked d_i^{MAX} he knows. For instance, it is impossible for him to even compare $d_{a,fair}$ to any of the d_i^{MAX} as only the LDS knows the secret scaling and shifting values used for the masking operation. Hence, even with the additional knowledge of the $d_{a,fair}$ and L_{fair}, the adversary cannot guess the value of k with a probability higher than a random guess. Thus, the advantage of the adversary is negligible in this case as well.

Considering the previous arguments, we have the following:

$$\begin{aligned} Adv_a^{IDT}(A) &= Pr(k' = k | L_{fair} = L_a) Pr(L_{fair} = L_a) \\ &\quad + Pr(k' = k | L_{fair} \neq L_a) Pr(L_{fair} \neq L_a) - 1/N \\ &= 1/N \cdot 1/(N+1) + 1/N \cdot N/(N+1) - 1/N = 1/N - 1/N \\ &= 0 \end{aligned}$$

Thanks to the independence of k' conditioned on the outcome L_{fair}. Thus, the identifiability-advantage is negligible.

User Coordinate-Linkability Advantage

Similarly to the identifiability advantage, there could only be two possible outcomes of any PPFRVP algorithm A, represented by the two cases $L_{fair} \neq L_a$ and $L_{fair} = L_a$. Hereafter we show that the advantage of the adversary is negligible in both cases.

1. $\mathbf{L_{fair} = L_a}$: In this case, the adversary does not learn any additional information about the coordinates of any two users j, k. As the masked and ordered distances cannot be linked to a specific coordinate with a success probability higher than 1/2, the adversary cannot guess whether the coordinate value b_j is larger or smaller than b_k with a probability higher than a random guess (1/2). In fact, as the order among the masked distances is a relative measure between locations that is position-independent, it does not provide any additional information about the values of the coordinates of L_j, L_k. Thus, the advantage of the adversary is negligible.

2. $\mathbf{L_{fair} \neq L_a}$: In this case, the adversary can once again compute the distance $d_{a,fair}$ between L_{fair} and L_a. As the distance by itself conveys no

information about the orientation or relative position between L_j and L_k, $\forall j, k \in \{1, \ldots, N\}$ and $j \neq k$, the adversary cannot guess whether the coordinate b, randomly chosen by the challenger, is larger or smaller for L_j with respect to L_k with a higher certainty than a random guess. Thus, his advantage is negligible.

Similarly to the identifiability advantage, we obtain:

$$\begin{aligned} Adv_a^{c-LNK}(A) &= Pr(r = 0 \wedge b_j \leq b_k | L_{fair} = L_a) Pr(L_{fair} = L_a) \\ &+ Pr(r = 0 \wedge b_j \leq b_k | L_{fair} \neq L_a) Pr(L_{fair} \neq L_a) \\ &+ Pr(r = 1 \wedge b_j > b_k | L_{fair} = L_a) Pr(L_{fair} = L_a) \\ &+ Pr(r = 1 \wedge b_j > b_k | L_{fair} \neq L_a) Pr(L_{fair} \neq L_a) \\ &- 1/2 \\ &= Pr(r = 0) \cdot Pr(b_j \leq b_k) + Pr(r = 1) \\ &\cdot Pr(b_j > b_k) - 1/2 = 1/4 + 1/4 - 1/2 = 0 \end{aligned}$$

Thanks to the independence of the coordinate b from the outcome L_{fair}. Thus, the coordinate-linkability is negligible.

User Distance-Linkability Advantage
The PPFRVP algorithm defined in this chapter takes as inputs the preferred rendez-vous locations L_i of each user $u_i \in \mathbb{U}$ and outputs both $f(L_fair)$ and the set of randomized (but order-preserving) maximum distances $d_i^{max}, \forall u_i \in \mathbb{U}$. By means of an example, we show that there is at least one case in which our PPFRVP algorithm does not satisfy distance-linkability.

Suppose that, at Step 3 of the distance-linkability game, $\mathbb{C}$ chooses a value $s > \max_{u_i \in \mathbb{U}} d_i^{max}$. At Step 4, u_a obtains (s, j, k) and it knows that s is larger than any of the maximum randomized distances that it already possesses. Moreover, u_a also knows that the order-preserving randomization procedure $Rand(.)$ is such that $d_i^{max} = Rand(c_i^{max}) > c_i^{max}$, i.e., the randomization strictly increases the output compared to the input because the two randomizing factors r_i, s_i are positive. Hence, if $s > \max_{u_i\ \mathbb{U}} d_i^{max}$, u_a knows that for sure $s > d_{j,k}, \forall j \neq k$. Thus, u_a can win the game with non-negligible probability by choosing $s^* = 0$, proving that in this case our PPFRVP algorithm A does not satisfy user distance-linkability. □

Active Adversary We consider three main categories of active attacks against PPFRVP protocols, namely (i) the collusion among users and/or LDS, (ii) the fake user generation and/or replay attacks and (iii) unfair rendez-vous location.

Collusion Regardless of the protocol used or the encryption methods, in the case when users collude among themselves the published fair result (together with the additional information malicious users may get from colluders) can be used to construct exclusion zones, based on the set of equations and known constraints.

An exclusion zone is a region that does not contain any location preferences, and the number of such exclusion zones increases with the number of colluders. We are currently working on quantifying this impact on our optimization and encryption methods. However, in the unlikely case of collusion between the LDS and the participants, the latter will be able to obtain other participants' preferences. In order to mitigate such a threat, the invited participants could agree on establishing a shared secret by using techniques from threshold cryptography [139]. The LDS should then collude with at least a given number of participants in order to obtain the shared secret and learn L_i.

Fake Users Generating fake users can be attempted both by the LDS and by any meeting participant, in order to disrupt or manipulate the computations of the fair rendez-vous location. However, the security of our algorithms prevents such attacks from succeeding. In case the LDS generates fake users, it would not be able to obtain the secret that is shared among the honest users and which is used to derive the secret key $K_s^{M_v}$ for each session v. This attack is more dangerous if a legitimate participant creates a fake, because the legitimate participant knows the shared secret. In this scenario, however, the LDS knows the list of meeting participants (as it computes the fair rendez-vous location) and therefore it would accept only messages digitally signed by each one of them. Here we rely on the fact that fake users will not be able to get their public keys signed by a CA. Replay attacks could be thwarted by adding and verifying an individually signed *nonce*, derived using the shared secret, in each user's message.

Unfair RV The last type of active attack could lead to the determination of an unfair rendez-vous location. Maliciously modifying or untruthfully reporting the maximum masked values (Step B.3 of Figure 3.5) could deceive the LDS to accept the false received index as the maximum value, and therefore potentially lead to the determination of a subfair rendez-vous location. However, this is rather unlikely to happen in practice. For instance, even if in Step B.3 a user falsely reports one of his values to be the maximum when actually it is not, this would cause the algorithm to select a subfair rendez-vous location if and only if no other user selected a smaller value as the maximum distance.

Complexity Analysis

Table 3.2 summarizes the complexity results for our two protocols, both for the client devices and for the LDS. As it can be seen, the client complexity is in general $O(N)$, where N is the number of users. However, there is a notable exception for the BGN-based scheme; the number of exponentiation required for a single decryption is $O(\sqrt{T})$ [24], where T is the order of the plaintext domain. In Section 3.5, we show how this charateristic impacts the decryption performance.

The LDS complexity for both protocols is in general $O(N^2)$, with the notable exception of BGN, where in addition to multiplications and exponentiations the

Table 3.2: Asymptotic complexity of the proposed PPFRVP protocols, where N is the number of participants. The *Distance* protocol is the one used in the module A of Figure 3.5, whereas PPFRVP includes modules A,B and C.

CLIENT	PROTOCOL	BGN *(mod n)*	ELGAMAL-PAILLIER *(mod n^2)*	LDS	BGN *(mod n)*	ELGAMAL-PAILLIER *(mod n^2)*
Mult.	Distance	O(1)	O(N)	**Mult. Exp.**	$O(N^2)$	$O(N^2)$
	PPFRVP					
Exp.	Distance	O(1)	O(N)	**Bilinear mapping**	$O(N^2)$	-------
	PPFRVP	*$O(N\sqrt{T})$*				
Memory	Distance	O(1)	O(N)	**Memory**	$O(N^2)$	$O(N^2)$
	PPFRVP	*O(N)*				
Comm.	Distance	O(1)	O(N)	**Comm.**	O(N)	$O(N^2)$
	PPFRVP	*O(N)*			*$O(N^2)$*	

schemes requires additional $O(N^2)$ bilinear mappings. These operations are required in order to support the multiplicative property of the BGN scheme. The bilinear mappings are not required in the ElGamal-Paillier scheme. Nevertheless, the exponentiation operations can be pre-computed in BGN (because the h value is public), as opposed to the ElGamal-Paillier-based approach.

3.5 Implementation and Performance Evaluation

In this section, we discuss the results of the performance measurements using our implementations of the proposed algorithms on Nokia devices.

The tests were conducted on Nokia N810 mobile devices (ARM 400 MHz CPU, 256 MB RAM). The operating system on the N810s is the Linux-based Maemo OS2008, and we wrote our applications using the Qt programming language, which is optimized for such OS. The LDS has been implemented on a 2 GHz Intel CPU with 3 GB RAM, running the Ubuntu 9.04 Linux. Figure 3.6 shows the software architecture diagram, where the function of each block is self-explanatory, whereas Figure 3.7 depicts the prototype application running on a Nokia N810.

For the elliptic curve BGN-based PPFRVP protocol, we measured the performance using both a 160-bit and a 256-bit secret key, whereas for the ElGamal-Paillier-based one we used 1024-bit secret keys. As BGN is an elliptic curve-based scheme, much shorter keys can be used compared to ElGamal and RSA. A 160-bit key in elliptic curve cryptosystems is generally believed to provide equivalent security as a 1024-bit key in RSA and ElGamal [135]. Readers should note that we measured only the computation time on the devices, without the message communication delays. As several wireless interfaces (WiFi, cellular, bluetooth)

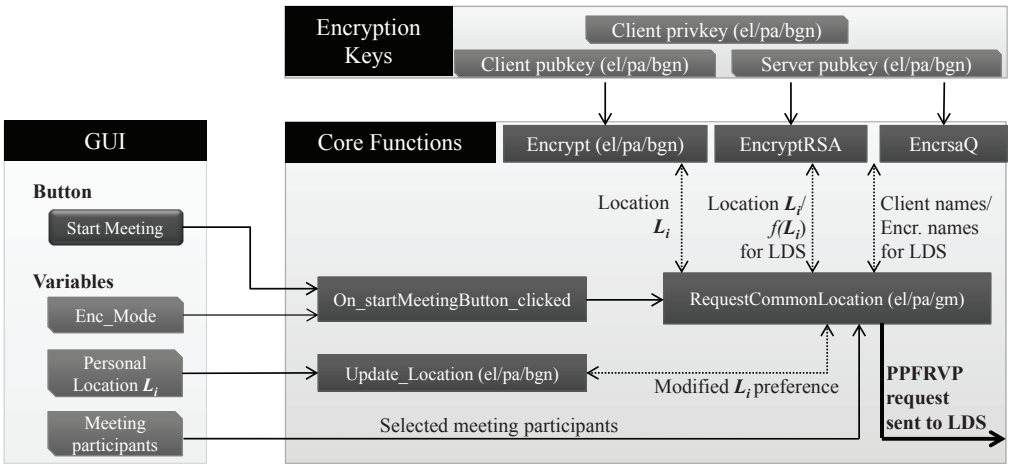


Figure 3.6: Architecture diagram of the client PPFRVP prototype application.

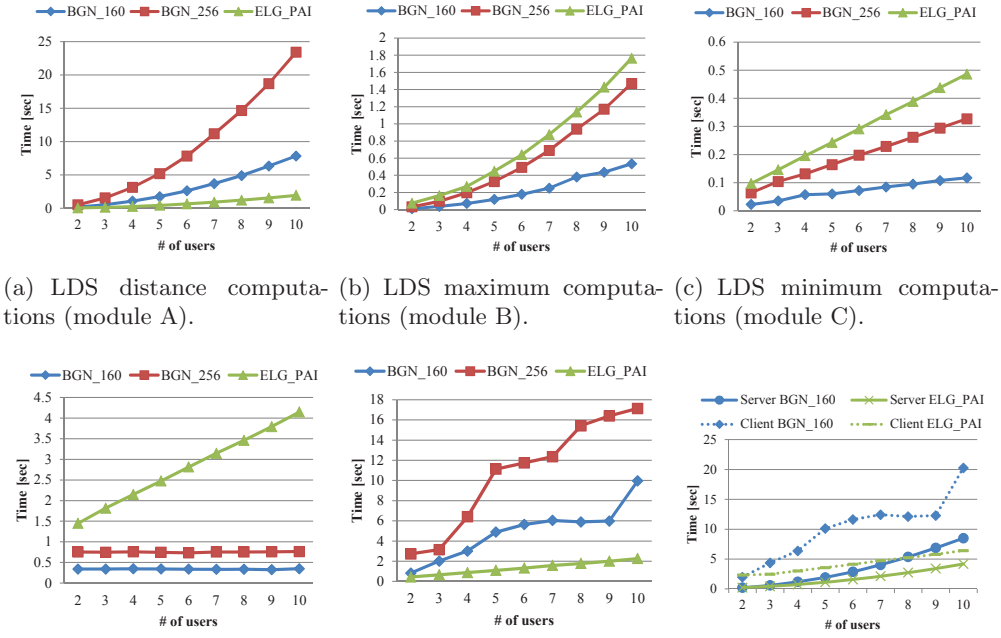
Figure 3.7: Prototype PPFRVP application running on a Nokia N810 mobile device. The image on the left is the main window, where users add the desired meeting participants. The image on the right is the map that shows the fair rendez-vous location (green pin) and the user-preferred rendez-vous location (red pin).

with different connection data rates and delays could be used to access these services, we do not measure time delays here.

LDS Performance

Figures 3.8(a), 3.8(b) and 3.8(c) show the computation time required by the LDS. We can see that it increases with the number of users, and that the ElGamal-Paillier algorithm is the most efficient across all computations, requiring 4 seconds to execute the PPFRVP protocol with 10 participants. The two BGN-based algorithms are less efficient, but are still practical enough (9 seconds). The CPU-intensive bilinear mappings in BGN are certainly one important reason for such delays.

Regarding the subsequent modules B and C of the PPFRVP protocol, we



(a) LDS distance computations (module A).

(b) LDS maximum computations (module B).

(c) LDS minimum computations (module C).

(d) Client distance computations (module A).

(e) Client max/argmin computations (module B/C).

(f) Total client and LDS run times (modules A+B+C).

Figure 3.8: Performance measurements.

observe a that the BGN-based algorithms outperform the ElGamal-Paillier one (Figure 3.8(b) and 3.8(c)). The maximum computations on the LDS require 0.5 seconds for the 160-bit BGN algorithm, whereas the ElGamal-Paillier takes almost 2 seconds. A similar result can be observed for the minimum computations. There are two main reasons for this. First, there are no bilinear mappings involved in these modules and second, the BGN-based algorithms use much smaller key sizes. From a practical perspective, both the ElGamal-Paillier and the BGN algorithms have good performance in modules B and C of the PPFRVP protocol.

Client Performance

Figures 3.8(d) and 3.8(e) show the different computation times on the Nokia N810 mobile device. As it can be seen, thanks to the efficient use of the homomorphic properties of our BGN-based algorithm, this protocol is the most efficient for the distance computations, requiring only 0.3 seconds, independently of the number of users. The alternative protocol, on the contrary, needs 4 seconds with 10 participants. However, in the subsequent phases, the results are not as good, as the BGN protocol makes intensive use of bilinear mappings.

Overall, we can see that the ElGamal-Paillier protocol has a better performance than the BGN-based one, both on the client and on the LDS. Nevertheless, both schemes are practical enough and have acceptable time requirements

in order to be implemented on current generations of mobile devices. Thanks to the homomorphic properties of BGN, the pairwise distances can be obliviously computed by the LDS, without involving any decryption/re-encryption operation from the clients (as opposed to the ElGamal-Paillier alternative). Even with a comparatively larger security resistance, the BGN scheme is still faster than the alternative one. It is also important to notice that the design of the proposed BGN distance algorithm allows it to perform well, independently of the number of participants.

3.5.1 Discussion

The implementation measurement results of our two PPFRVP algorithms show that the ElGamal-Paillier based algorithm has a better overall performance than the BGN-based alternative. On a comparable security level and without considering communication delays, both the LDS and the client device require less computation time for the former algorithm compared to the latter. However, aside from the performance, the BGN-based algorithm presents several advantages. First, it involves three less message exchanges between each client and the LDS (Figure 3.3) compared to the alternative algorithm. Second, much shorter security parameters can be used in order to achieve the same resistance to attacks, and thus lower the memory requirements on the client devices. Third, malicious users cannot change the masked user-preferred meeting-location coordinates once they have been sent to the LDS, as there are no decryption/re-encryption operations in the BGN-based algorithm, as opposed to the ElGamal-Paillier based one.

3.6 User Study

In this section, we present the methodology and results of the user study we conducted with our prototype application.

Novel LSB services, such as deals and check-ins, are offered by large service providers such as Google and Facebook. In order to assess users' opinions about the potential and challenges of such services, we conducted a targeted user study on 35 respondents, sampling a population of technology-savvy college students (in the age group of 20-30 years) and non-scientific personnel. The questionnaires are based on the privacy and usability guidelines from [33, 98].

The entire study consisted of three phases: the purpose of Phase 1, during which respondents answered a first set of 22 questions without knowing the subject of the study, was to assess the participants' level of adoption of mobile LSBS and their sensitivity to privacy issues in such services. The answers to these questions are either "Yes" or "No", or on a 4-point Lickert scale (where 1 means *Disagree*, 4 is *Agree*). In Phase 2, the respondents were instructed to use our pro-

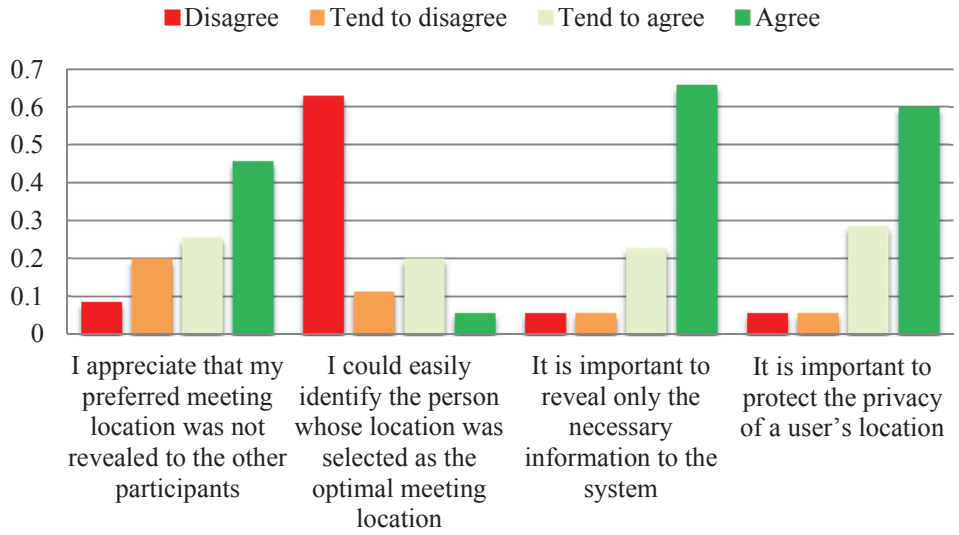


Figure 3.9: Summary of the user-study results for Phase 3.

totype mobile FRVP application. Finally, in Phase 3, the participants answered the second set of 12 questions, choosing from a 4-point Lickert scale, after having used our application. The purpose of this phase was to capture users' opinions on the usability of our application, and to assess whether privacy undermines usability or performance in such an application.

Phase 1 The majority of the participants are males, 20-25 years old. Around 86% of them use social networks, and 74% browse the Internet with a mobile device. When organizing meetings, 54% of the time they involve groups of 4 people and 29% groups of 6 individuals, and participants use their mobile device for organizing 63% of such meetings. Although only 14% are aware of existing LSBSs, 51% would be *very* or *quite* interested in using a LSBS such as the FRVP. However, they are sensitive to privacy (98%) and anonymity (74%) in their online interactions, especially with respect to the potential misuse of their private information by non-specified third-parties (88%). Furthermore, most of the participants (80%) agree that no organization or person should disseminate their personal information without their knowledge.

These results indicate that, although rare at the moment, LSBSs are perceived as interesting by the majority of the sampled population, who are also the most likely to adopt LBS technologies [111]. With respect to privacy, they agree that it is crucial for the acceptability of such services, and thus LSBSs should work properly by requiring a minimum amount of personal information.

Phase 2 In this phase, the participants were instructed on the usage of our PPFRVP prototype application and they used it several times to determine a fair rendez-vous point. The participants were seated in separate places of the

experimentation area and they could not speak to other participants or see their device screens.

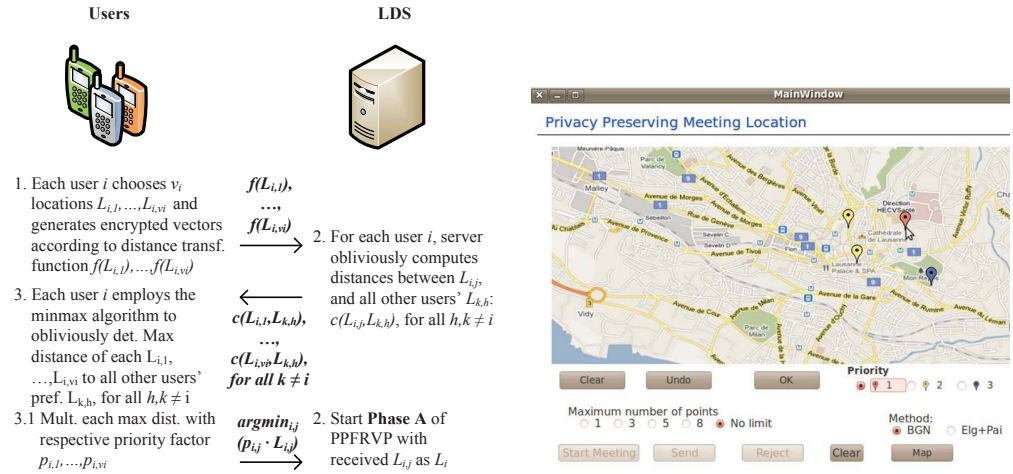
Phase 3 Figure 3.9 summarizes the main findings of this phase. After using our application, all participants tend to agree (34%) or agree (66%) that it was easy to use, and that they could quickly compute the task (97%). Around 80% feel that it was easy to learn to use such application, and 91% tend to or agree that the GUI was clearly organized. More than 68% agree that the application was interesting to use, and if we include those who *tend to agree* as well, all but one participant found it interesting. With respect to privacy in such applications, 66% agree that it is important to reveal only the necessary information to the system, in order to compute the fair rendez-vous location. More than 71% appreciated that their preferred rendez-vous point was not revealed to other participants, and only 8% did not care about the privacy of their rendez-vous location preference. Only 26% of the participants were able to identify to whom the FRVP location belonged to, which was to be expected. The users ran our application in groups of 5 during the experimentation, and therefore there was always one person out of five that knew that the FRVP location was his preferred location.

From a software developer's standpoint, this means that both ease of use and privacy need to be taken into account from the beginning of the application development. In particular, the privacy mechanisms should be implemented in a way that does not significantly affect the usability or performance. Moreover, the acceptance of LSBS applications is highly influenced by the availability of effective and intuitive privacy features.

3.7 Extension

The PPFRVP protocol defined in Section 3.3 allows each user i to select one preferred location L_i in order to determine the fair rendez-vous location L_{fair}. A natural extension of the existing protocol would consist in allowing any user i to select multiple preferred locations $L_{i,1}, \ldots, L_{i,vi}$, and to associate a personal priority to any such preference. This way, the users would have more flexibility in making choices and the output of the PPFRVP algorithm would better reflect the ranking of the location preferences. Moreover, priorities could also be used to assign weights to location preferences based on their availability and accessibility, if such information is available. Hereafter, we propose such an extension to our PPFRVP protocol and discuss its characteristics. Figures 3.10(a) and 3.10(b) show the new protocol and its preliminary implementation, respectively.

The multi-point preferences are considered by assigning a priority to each preferred location $L_{i,j}$ for all users i and preferences j. One way to include them in the distance computations is to assign weighting coefficients $p_{i,j}$ for the maximum distances $c(L_{i,j}, L_{k,h})$ computed at the end of Step 3 in Figure 3.10(a);



(a) Extension of the existing PPFRVP protocol. (b) Prototype application.

Figure 3.10: Extension of PPFRVP to multiple user-preferred locations.

this way, the highest priority could be defined by using the lowest value of $p_{i,1} = 1$, whereas the lower priorities could be assigned higher values of $p_{i,2} = p_{i,3} = 2$ (as in Figure 3.10(b)). As a result, the minimum of these maximum meta-distance is crucial for each client in order to select his own prioritized location in Step 3.1, which will then be sent to the LDS for the continuation of the PPFRVP computations.

3.7.1 Privacy Discussion of the Multi-Point Extension

The proposed extension to the PPFRVP protocol relies on the same cryptographic primitives as the original PPFRVP protocol. As the operations performed by the LDS are the same (oblivious distance computations and randomization), the extended PPFRVP protocol also inherits the privacy guarantees possessed by the single-location PPFRVP one. Therefore, the PPFRVP protocol (BGN- and ElGamal-Paillier-based) with the extension to multiple user-preferred locations preserves identifiability and coordinate-linkability privacy in the passive adversary scenario. However, it retains the same vulnerabilities in the active adversary scenario.

3.7.2 Performance Discussion of the Multi-Point Extension

With respect to the ElGamal-Paillier-based protocol the BGN-based distance computations of the extension clearly reduce the number of message exchanges between each client and the LDS. However, as there is a decryption operation performed by the clients in Step 3, the distance computation for the extended protocol with the BGN scheme would put an extra complexity factor, increas-

ing the overall complexity of the BGN-based extended PPFRVP protocol to $O(N\sqrt{T}\max_i vi)$, where vi is the number of location preferences of user i, compared to $O(N\max_i vi)$ of the ElGamal-Paillier-based scheme. The communication complexity would however remain the same for both protocols, which is $O(N\max_i vi)$. Hence, the ElGamal-Paillier-based, extended PPFRVP protocol would be more preferable from a performance standpoint.

3.8 Related Work

Hereafter, we present some works in the literature that address, without protecting privacy, strategies to determine the optimal meeting location. To the best of our knowledge, ours is the first work to address the problem in a privacy-preserving way, by determining one location among a set of proposed locations.

Santos and Vaughn [138] present a survey of existing literature on meeting-location algorithms and propose a more comprehensive solution for such a problem. Although considering aspects such as user preferences and constraints, their work (or the surveyed papers) does not address any security or privacy issues. Similarly, Berger et. al [14] propose an efficient meeting-location algorithm that considers the time in-between two consecutive meetings. However, all private information about users is public.

In the domain of Secure Multiparty Computation (SMC), several authors have addressed privacy issues related to the computation of the distance between two routes [63] or points [99, 145]. Frikken and Atallah [63] propose SMC protocols for securely computing the distance between a point and a line segment, the distance between two moving points and the distance between two line segments. Zhong et al. [162] design and implement three distributed privacy-preserving protocols for nearby friend discovery, and they show how to cryptographically compute the distance between a pair of users. However, due to the fully distributed nature of the aforementioned approaches, the computational and communication complexities increase significantly with the size of the participants and inputs. Moreover, all parties involved in the computations need to be online and synchronized.

There have also been several research results in the literature that focus on the problem of privacy-preserving *location-based queries* and *location sharing* or anonymous *location check-ins*. However, these research efforts attempt to solve issues that are orthogonal, and uniquely different, from the ones addressed in this paper. Jaiswal and Nandi [91] propose a privacy-preserving platform, called *Trust No One*, for privately locating nearby points-of-interest. Their architecture relies on three non-colluding parties, i.e., the mobile operator, the LBS provider, and the matching service, for decoupling user locations from user queries. The architecture proposed by Jaiswal and Nandi [91] addresses the problem of location-privacy preserving information retrieval, which is different from our focus. In

another related effort, Olumofin et al. [122] use *Private Information Retrieval (PIR)* techniques to preserve user's location privacy in location-based queries, e.g., retrieving points-of-interest information related to current location. This line of work focuses on private information retrieval based on user locations, whereas we focus solely on private computations based on user provided locations or location preferences. Besides these, many earlier research efforts [64, 6, 160] on anonymizing location-based queries employed the k-anonymity approach. In this approach, for each query issued by a user, a trusted third party generates an *obfuscation* or cloaking region containing $k-1$ other users. The third party then forwards the cloaking region, instead of the user's actual location, in the user query to the service provider. In our work, the goal is not to simply anonymize users' location data, but to hide this data from the service provider and to enable provider-oblivious computations on it.

In the direction of anonymous location sharing, Pidcock et al. [128] propose a novel architecture called *ZeroSquare* where the main goal is to provide a location hub for privacy-preserving geospatial applications. The main idea of the authors is to decouple user (profile) information from location information by assuming two non-colluding entities that store this information. Their work is different from ours because they do not consider the problem of privately computing some function based on the location data, rather they want to enable privacy-preserving location sharing in mobile applications. Contrary to the work by Pidcock et al. [128], Guha et al. [76] propose a privacy-preserving system that allows users to set location-triggered alarms based on presence at specific locations, rather than sharing location coordinates. In another related effort, Herrmann et al. [80] propose two constructions for privacy-preserving location sharing by using broadcast encryption techniques. Their construction prevents the service provider from learning the identity of the user and her location, but allows it to determine which other users are allowed to know the user's location. Contrary to our work, which focuses on efficient computation of some common function based on shared user locations while keeping these locations private (from the service provider), they focus on keeping both the identities and the shared locations private from the service provider. Carbunar et al. [30] also propose a set of privacy-preserving protocols, using well-known cryptographic constructs, which anonymously proves to a venue that a user checked-in (her location) a certain number of times. In our work, we do not consider the problem of anonymously proving presence at a particular location. Rather, our goal is to hide the user's location claim (or location preference) from other users and the service provider.

3.9 Summary

Novel LBS applications are taking advantage of the numerous technological advances in order to provide a better and more efficient user experience. Major online providers are leveraging on the enthusiasm for such services in order to deliver increasingly sophisticated location-based applications, such as location-sharing-based (LSB) applications. However, the existing users' concerns about privacy in such services need to be effectively considered from the beginning, as they are essential for the success and adoption of such services.

In this chapter, we have addressed the problem of privacy in the FRVP problem by providing practical and effective solutions based on homomorphic properties of well-known cryptosystems. We have designed, implemented on real mobile devices and evaluated the performance of our privacy-preserving protocols for solving the fair rendez-vous problem in a privacy-preserving manner. Our solutions are proved to be effective in terms of privacy, have acceptable performance, and do not create additional overhead for the users. Moreover, we have extended the proposed solutions to include more realistic scenarios where users have several preferred and prioritized locations preferences. Finally, our user-study showed that the proposed privacy features are crucial for the adoption of any such application, which reinforces the need for further exploration in privacy of LSB services.

Publications: [20, 19]

Part II

Network-Layer Privacy Exposure

Chapter 4

Privacy of Social Communities in Pervasive Networks

Wireless network operators increasingly deploy WiFi hotspots and low-power, low-range base stations in order to satisfy users' growing demands for context-aware services and performance. In addition to providing better service, such capillary infrastructure deployment threatens users' privacy with respect to their social relationships and communities, as it allows infrastructure owners to infer users' daily social encounters with increasing accuracy, much to the detriment of their privacy. Yet, to date, there are no evaluations of the privacy of communities in pervasive wireless networks. In this chapter, we address the important issue of privacy in pervasive communities by experimentally evaluating the accuracy of an adversary-owned set of wireless sniffing stations in reconstructing the communities of mobile users. During a four-month trial, 80 participants carried mobile devices and were eavesdropped on by an adversarial wireless mesh network on a university campus. To the best of our knowledge, this is the first study that focuses on the privacy of communities in a deployed pervasive network and that provides important empirical evidence on the accuracy and feasibility of community tracking in such networks.

Chapter Outline In Section 4.1, we introduce the privacy issue in pervasive social networks, and in Section 4.2 we detail the experimental trial framework together with its system and network models. In Section 4.3, we outline the community and privacy analysis. In Section 4.4 we present the results of the analysis of communities and their privacy *vis-à-vis* the external adversary. We present the related work in Section 4.5, and we summarize the chapter in Section 4.6.

4.1 Introduction

Every day, mobile operators collect large amounts of users' data that is mined for commercial and performance goals, such as billing, throughput, coverage and usage statistics. In addition to the explicit information (such as cost, duration, location) that can be derived from the communications, operators and infrastructure owners are able to gain additional knowledge based on the communication and contextual patterns, without any action from the user for this regard [75, 81]. Users' home/work locations [75, 81], activities [107], interests [120] and social networks [38, 105] can be inferred from their location and social interactions, much to the detriment of not only their own privacy, but also to that of their peers.

More recently, telecom manufacturers have also added support for seamless, low-cost, wireless device-to-device communications, such as Nokia Instant Community [36], AirDrop by Apple [4] and FlashlinQ by Qualcomm [37], thus complementing existing infrastructure-based communications. The possibility of real-time data sharing among devices, without the need for infrastructure, enables people to form localized and short-lived groups or *communities* of users, which can emerge in scenarios where the infrastructure is inadequate, expensive, untrusted or hostile [132, 55]. Although still an emerging research subject in the wireless domain [161], pervasive communities and their structured networks of interactions are able to significantly improve the performance of opportunistic networks [85, 31], by leveraging on the structural properties and patterns of the evolving user interactions. In the literature, there are several routing and packet-forwarding algorithms [83, 86, 31] that exploit the underlying evolving social interactions to improve the network performance, mostly based on the frequency of recorded Bluetooth encounters. Similarly, social communities have been studied from the behavioral perspective [38, 48, 72], in order to analyze people's preferences and group formation characteristics. The undisputed value of friendship networks and social ties to service providers such as Facebook and Twitter has also dramatically increased their monetary value [28], as more and more targeted advertisements and tailored services are proposed to groups of users with similar attitudes and interests.

In spite of the soaring interest for the analysis and exploitation of pervasive communities in the wireless domain, in terms of privacy very little has been achieved. Privacy of communities and their members is a major concern in regions where the ability to keep such information from being inferred by unscrupulous third-party providers or suppressive governments is critical [132, 55]. Even in developed countries, where each person has (on average) more than one mobile subscription [88] and where small and pervasive base stations (femtocells) already outnumber conventional base stations [54], the issue of community privacy is extremely relevant. Furthermore, the increased availability of public WiFi hotspots and the rapid deployment of low-power and low-range cellular base sta-

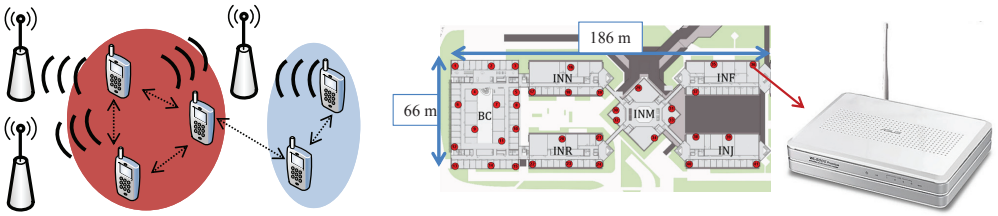
tions (femtocells) [54] makes such inference even more accurate, as more precise user proximity data can be collected, regardless of the kind of upper-layer protocols and applications. The risks of unsolicited user profiling, data censorship, racial discrimination and political repression, based on users' physical proximity derived from short-range communications, are a major concern. Because most of the existing literature on communities in wireless networks has been primarily focused on performance or human behavior, to the best of our knowledge there is no single empirical work that has addressed the issue of the privacy of communities in deployed wireless networks.

In this chapter, we address the problem of community privacy by taking a comparative analysis of the exposure of social relationships and encounters in a deployed wireless peer-to-peer (P2P) network. Over a four-month trial (March-June 2011) with 80 participants, we studied and quantified the extent of leakage of private community information by users, by providing empirical evidence about the network or infrastructure owner's accuracy of reconstruction of the social communities of people. Our work is unique in three respects:

- We provide the first privacy analysis of the exposure of community information in a deployed wireless network.
- We experimentally evaluate and compare the wireless sniffing stations owner's accuracy of reconstruction of the social communities of people, based on the observed traffic patterns, with the local proximity and the encounter data that is collected by the mobile devices.
- We characterize the evolution of the social interactions among the participants, and we evaluate the strength of their interactions by implementing three different social interaction measures that take into account the number, the proximity, the recency and aging effects of social relationships in the underlying wireless network.

4.2 System Architecture

During four months (March-June 2011), we conducted a large-scale trial with 80 participants on the EPFL university campus, in order to collect encounter and proximity data. Similarly to previous data collection campaigns [71, 78, 47], we programmed and distributed 80 Nokia N900 smartphones to the volunteering participants, sampling a coherent population of master's students and instructors of two classes taught during the spring semester. The participants were asked to carry their device with them as frequently as possible, and they were allowed to use it as their primary phone. At the end of the trial, we obtained useful information from 66 devices, amounting to almost ten GBs of collected log data and over 8 million packets captured by the adversarial network. The remaining



(a) Trial network architecture.

(b) Deployed wireless mesh network of 37 APs controlled by the adversary.

Figure 4.1: Trial setup and deployed eavesdropping network controlled by the adversary.

14 devices were either not used regularly or did not collect the data properly, hence they were excluded from the analysis.

4.2.1 Device Configuration

The Nokia devices were configured with both standard infrastructure-based communications, such as cellular and WiFi, as well as with a novel WiFi-based P2P technology, called *Nokia Instant Community* or NIC [36]. Users could connect to both standard Internet services using the WLAN or cellular interface of the device, as well as to an experimental context-aware wireless P2P messaging platform − in order to exchange information with their physical neighbors in a P2P fashion (Figure 4.1(a)). Moreover, several campus and course-related applications were developed in order to stimulate and encourage the usage of the devices throughout the duration of the trial. In order to enhance the context-awareness of the pre-installed applications, the devices were running background services that collected and stored, at regular intervals of [1-30] seconds, information such as the list of neighbors, the associated Received Signal Strength Indicator (RSSI) and the time stamp in the local memory. Whenever a participant connected to the Internet with the device, the new encounter logs were uploaded on a centralized database storing all device logs. To preserve users' anonymity, we removed all personal identifier information (such as the mapping between MAC address - IMEI - participant ID) from the database.

4.2.2 Adversarial Model and Infrastructure

We emulate a practical adversary who monitors a fixed area using a limited number of wireless sniffing stations. Specifically, the adversary is the owner of a deployed wireless mesh network of 37 APs (Asus WL-500gP APs running OpenWRT Linux) in a specific region of the campus [8], covering one level of six interconnected buildings which have a very high user (student) density (Figure 4.1(b)). The coverage area includes the classrooms in which the two classes that

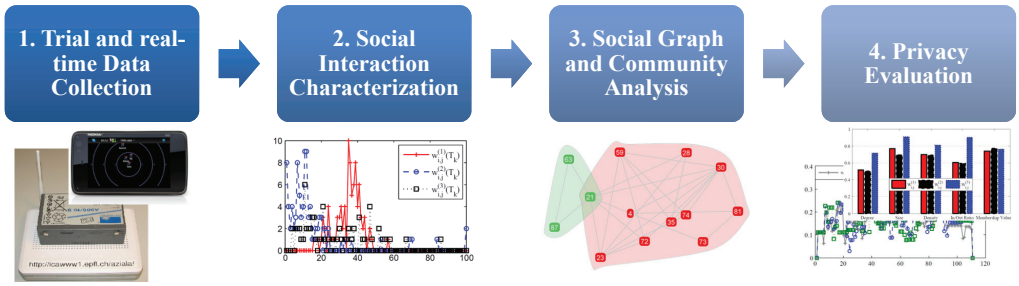


Figure 4.2: Flowchart of the pervasive community privacy evaluation process.

the students attended took place. We assume that the adversary passively eavesdrops on the participants' communications, and that he[1] periodically uploads the eavesdropped data to a centralized server, populating a unified log database for each AP.

In order to perform the pervasive community reconstruction attack discussed in the following section, we assume that the adversary collects the 3-tuple (Time stamp, Source MAC, RSSI) from the messages sent by the participants' smartphones. As encryption is sometimes used to protect the confidentiality of network and application-layer data in real networks, we assume that the adversary does not have access to such data. This reinforces the practicality and better embodies real-world limitations that an external adversary might have, being much weaker than the omniscient Dolev-Yao adversary [44]. Moreover, the information collected by the adversary is present in almost all kinds of wireless networks and technologies (such as Bluetooth, WiFi and cellular), which enlarges the applicability and scope of the results. In this work, we assume that the adversary does not have direct access to any information stored on the mobile devices, and that all devices are honest (i.e., not colluding with the adversary). As part of our future work, we will consider a stronger adversary that can collude and gain access to some of the mobile devices as well.

4.3 Community Analysis

In order to evaluate the extent of community information leakage in our setting, we first need to define the analytical framework that captures the pervasive community information from the collected data. In this section, we introduce some background on communities in wireless networks and describe how we evaluate communities and their privacy in our trial. A flowchart of the entire process is depicted in Figure 4.2.

[1]For conciseness and without loss of generality, we refer to the adversary in the masculine form, although both masculine and feminine forms apply.

4.3.1 Background

In society, people tend to organize themselves in social groups or communities, such as family, work colleagues and hobby groups, where members usually have stronger similarity traits with other members than with non-members [57]. From a graph-theoretic perspective, people and their relationships can be represented by an undirected graph $G = (V, E, W)$, where the vertex set V corresponds to people, the edge set E expresses the existence of a relationship between people, and the weight function W quantifies the intensity of such relationship. In their simplest form, communities can then be represented as subgraphs $\{C_i = (V_i, E_i, W_i,)\}_{i=1}^{M}$, where $C_i \subseteq G$ and M is the number of communities C_i.

Several community detection (or clustering) algorithms are present in the literature, and they work on either unweighted/weighted and undirected/directed graphs. Although hierarchical clustering [77] and modularity-based algorithms [115] − surveyed in [57] − have been applied to community detection, most of them lack a fundamental characteristic that is intrinsic to social communities. People are often members of several communities at the same time, such as friends, family members and work colleagues, and most of the aforementioned algorithms assign a single vertex to only one community. In order to allow a vertex to be assigned to multiple (possibly overlapping) communities, Palla *et al.* [126] developed a technique, the *Clique Percolation Method (CPM)*, which allows different communities to share vertices. The idea is that communities are formed by the union of adjacent k-cliques (complete graphs with k vertices), where two k-cliques are adjacent if they share $k-1$ vertices. Due to the social nature of our trial and the experimental setting, we use the CPM algorithm to detect pervasive communities based on physical proximity and encounter data.

After the pervasive communities have been discovered, several privacy-sensitive statistics can be obtained from the community structure, their overlap and their members. We describe the relevant statistics in Section 4.3.3.

4.3.2 Trial Framework

In order to model the collected encounter data using a graph, hereafter we describe the type of information that is used in order to define the existence and intensity of relationships between users.

Trial Data

In our trial, we have two sources of proximity information: (i) the local device logs collected by the mobile devices and containing encounter (list of neighbors, the time stamps and the RSSI values of received packets), and (ii) the adversarial (sniffing) logs containing the headers of the packets sent by the mobile devices,

which include the time stamps and RSSI values of received packets at the sniffing stations, as well as the device ID of the sender.

We use these two data sources in order to formulate the "strength" or intensity of the social relationships between users and to define the weights of the edges connecting the respective vertices in the social graph $G = (V, E, W)$. There are two types of proximity information in our network: device-to-device RSSI data (collected on the devices) and device-to-AP RSSI data (collected by the adversary). From the local device logs, we can directly obtain the device-to-device proximity information because the recorded RSSI values on the receiving device depend on the real distance to the sending device. However, this is not exactly the case for the RSSI values recorded by the adversarial network, as they depend on the distance between the sending device and the receiving sniffing station, and not the receiving mobile device. Therefore, the adversary needs to derive the device-to-device proximity information from the device-to-AP RSSI values. Hence, we first need to estimate the position of a device, and then compute the device-to-device proximity information in order to determine the weights between vertices of the social graph.

To this end, we developed a robust localization algorithm based on RSSI trilateration [23], which determines the estimated position of a received packet based on the RSSI at all sniffing stations that received that packet. Using the position estimate, we then compute the distance and RSSI between mobile devices, as described later in this section.

Social Interaction Intensity

We define three distinct weight functions $\{w_{i,j}^{(d)}\}_{d=1}^{3}$ between the vertices $i, j \in V$, taking progressively into account the proximity, the intensity and the aging and recency of the relationships between users. We divide the timeline of the trial into discrete time intervals $\{T_k\}_{k=1}^{N}$, where $N = 120$ days, and for each day T_k we define the weights $w_{i,j}(T_k)^{(d)}$ between users i, j.

The first and simplest weight function is the (shifted, non-negative) average of the RSSI value between a pair of users i, j for each day T_k, defined as

$$w_{i,j}(T_k)^{(1)} = \left(\frac{1}{c_{i,j}(T_k)} \cdot \sum_{q=1}^{c_{i,j}(T_k)} RSSI_{i,j}(T_k, q) \right) - r_{min}$$

where $c_{i,j}(T_k)$ is the sum of the number of packets received by i (and sent by j) and received by j (and sent by i) during the day T_k, $RSSI_{i,j}(T_k, q)$ is the RSSI value of a packet q received by a user i (and sent by j) or received by j (and sent by i) during the day T_k, and r_{min} is the minimum RSSI value that was recorded during the trial. For instance, we fix $r_{min} = -100$ dBm as no RSSI values lower than -100 dBm have been recorded by any device. Apart from the intensity, this

weight function does not consider the duration of the encounters (as it normalizes the intensity by the number of packets) between users or any aging or recency effect.

The second weight function takes into account the duration of the encounters through the sum of the (shifted, non-negative) RSSI values between users i, j, for each day T_k. It is defined as

$$w_{i,j}(T_k)^{(2)} = c_{i,j}(T_k) \cdot w_{i,j}(T_k)^{(1)} = \sum_{q=1}^{c_{i,j}(T_k)} (RSSI_{i,j}(T_k, q) - r_{min})$$

As the devices who are in continuous radio contact automatically exchange more context messages than the non-connected devices, this weight function takes into account the duration of the contacts, in addition to their intensity.

As communities of mobile devices are dynamic and evolve over time, the third weight function captures the natural evolution of social relationships between individuals, where past experience, recency and current state determine the intensity of interactions among people [125]. In this way, two users that have spent much time together in the past, but have not met on a given day, would still keep a relationship during that day (which is not the case for $w_{i,j}(T_k)^{(1)}$ and $w_{i,j}(T_k)^{(2)}$), even if its intensity is lower due to the aging effect − thus avoiding strong temporal fluctuations. Inspired by the formulations in [125, 157], we define the third weight function as

$$\begin{aligned} w_{i,j}(T_k)^{(3)} = \mathbf{1}_{c_{i,j}(T_k)>0} \left(\tau \cdot w_{i,j}^{(3)}(T_{k-1}) + (1-\tau) \cdot \gamma_{i,j}(T_k)\right) \\ + (1 - \mathbf{1}_{c_{i,j}(T_k)>0}) \cdot \left(w_{i,j}^{(3)}(T_e) \cdot \theta_{i,j}(T_k, T_e)\right) \end{aligned} \tag{4.1}$$

where

$$\gamma_{i,j}(T_k) = \frac{1}{\alpha} \cdot w_{i,j}(T_k)^{(2)}$$

$$\theta_{i,j}(T_k, T_e) = \exp\left(-\frac{\lambda(T_k - T_e)}{1 + \sum_{r=0}^{\min(T_k - T_e, T_e)} m_{i,j}(T_e - r)}\right)$$

$$m_{i,j}(T_k) = \begin{cases} 1 & \text{if } \gamma_{i,j}(T_k) > \beta \\ 0 & \text{otherwise} \end{cases}$$

and $\mathbf{1}_{c_{i,j}>0}$ is the indicator function, $0 \leq \tau \leq 1$ is the aging coefficient, $\alpha > 0$ is the normalization factor, $0 < \lambda \leq 1$ is the temporal decay value, $0 \leq T_e \leq T_{k-1}$ is the last day before T_k when users i, j exchanged messages, $m_{i,j}(T_k) \in \{0, 1\}$ is the recency factor that indicates whether a meeting took place during T_k or not, and $\beta \geq 0$ is the meeting threshold value. The idea behind the formulation is the following: If users i, j exchanged at least one message on a day T_k, then the weight of their edge is an exponential moving average of the aged weight − accumulated up to the day before (T_{k-1}) − and the recent day's weight; on

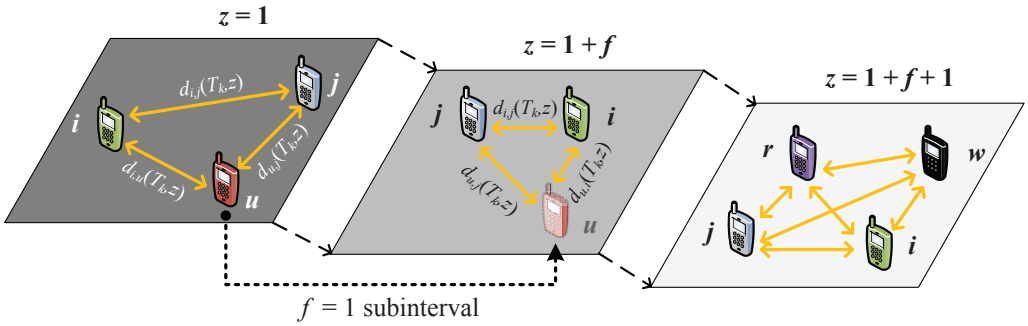


Figure 4.3: Users' positions estimates by the adversary. In this example, the adversary has the position estimate of user u at $z = 1$ but not at subsequent subintervals. In this case, u's last position estimate (at $z = 1$) is assumed to be valid in f subsequent subintervals. Here $f = 1$.

the contrary, if i, j did not exchange any message on day T_k, the current day's weight is a function of the previously accumulated weight, the frequency of their encounters just before the last encounter and the amount of time between the last time i, j had exchanged messages (T_e) and the current day T_k.

The weight functions can be directly applied to the local-device proximity information, as the available proximity information (time stamps, RSSI values from neighboring devices and their IDs) are sufficient for their computations. However, an intermediate step is required in order to compute the weights by using the external (adversarial) proximity information (time stamps, RSSI values from devices to sniffing APs and device IDs). In the following we show how to use the external proximity information in order to compute the edge weights.

User-Distance Estimation by the Adversary

As the adversary does not have access to device-to-device proximity data, he can decide to only use the estimated positions of a user i in a day T_k, defined as $P_i(T_k) = \{p_i(T_k, 1), \ldots, p_i(T_k, b)\}$, where b is the number of subintervals of a day T_k and $p_i(T_k, z) = (x_i(T_k, z), y_i(T_k, z)) \in \mathbb{R}^2$ is the estimated position of user i in the subinterval z of day T_k. Moreover, because there is a possibility that a user's packet may not be detected in each subinterval z, due to mobility or radio interference, we assume that the last position estimate $p_i(T_k, z_{last})$ of a user i is valid in f subsequent subintervals, if no $\{p_i(T_k, z_{last} + 1), \ldots, p_i(T_k, z_{last} + f)\}$ are available (Figure 4.3).

With such information, the adversary computes the edge weights as follows:

(1) $\forall z \in \{1, \ldots, b\}$, compute $p_i(T_k, z)$ for all users i observed on day T_k.

(2) $\forall z \in \{1, \ldots, b\}$, compute the estimated Euclidian distance $d_{i,j}(T_k, z) = ||p_i(T_k, z) - p_j(T_k, z)||$ between any two users i, j observed on day T_k.

(3) $\forall z \in \{1, \ldots, b\}$, compute the estimated RSSI value according to the adapted Haka-Okumura model for indoor radio propagation [26]

$$\widehat{RSSI}_{i,j}(T_k, z)[dBm] = P_t + 20 \log\left(\frac{\lambda}{4\pi}\right) + 10n \log\left(\frac{1}{d_{i,j}(T_k, z)}\right)$$

where $P_t = 20$ [dBm] is the transmission power of the mobile device, $\lambda = 0.125$ [m] is the wavelength, $n = 4.8$ is the path-loss exponent suited for office environments such as the university buildings under observation. The $\widehat{RSSI}_{i,j}(T_k, z)$ value replaces $RSSI_{i,j}(T_k, q)$ in the weight functions $w_{i,j}^{(d)}(T_k)$, where $z \in \{1, \ldots, b\}$.

Weight Distributions

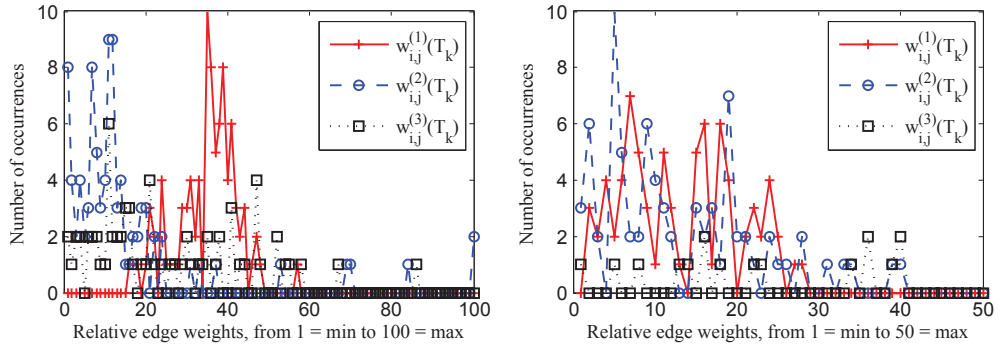
Due to the different features of a social relationship that each weight function models, their numeric values fall in different domains. For example, if $\alpha = 100$, $\beta = 1$, $\lambda = \tau = 0.5$ we have $0 \leq w_{i,j}^{(1)} < r_{min}$, $0 \leq w_{i,j}^{(2)} < 2.5 \cdot 10^5$ and $0 \leq w_{i,j}^{(3)} < 600$. It is therefore necessary to put them on the same scale for the identification of communities, as simply comparing the absolute values of the three weight functions is pointless. Hence, rather than comparing absolute values, we compare the weight distributions relative to the maximum of each weight function for each day T_k. To this end, we select an equal number of bins $I^{(d)}$ for each weight function $w_{i,j}^{(d)}(T_k)$. We then count the number of weight values that fall inside each such bin for all weight types, and we compare the distributions.

Figure 4.4(a) and Figure 4.4(b) show the relative edge weight distribution for a day T_k, by using the internal (local device) and external (adversarial estimate) input data, respectively. We see that, compared to the adversarial data, the local device data yields more pronounced characteristics for all three weight types and provides a more discriminating information set for the subsequent community detection phase, whereas the external data is less feature-rich due to the presence of uncertainty in the estimates of the proximity between users. This means that the adversary will likely struggle to infer with high accuracy the community characteristics for that day. We quantify such inaccuracies in Section 4.4.

Next, we describe the method we adopted to evaluate the extent of community information leakage and the related privacy measures.

4.3.3 Communities and Privacy

Having quantified the social interaction intensity as edge weights between any two trial participants, we now outline the community detection process, the suitable community statistics and privacy measures used to evaluate community privacy in our work.



(a) Using internal (local device) input data. (b) Using external (adversarial estimates) input data.

Figure 4.4: Relative edge weight distribution for different input data sets.

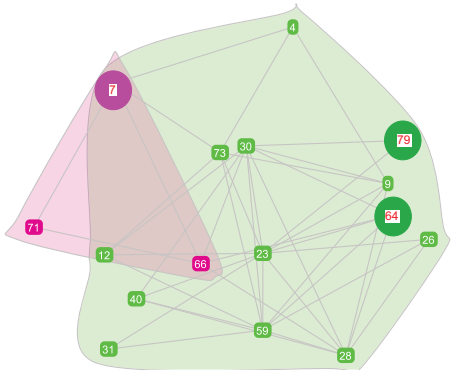
Community Detection

In its simplest form, the CPM community detection algorithm is defined for undirected and unweighted graphs [125], thus requiring only connectivity between vertices in order to discover communities. However, in order to consider the "strength" of the interactions between vertices, it was extended to work on weighted graphs by the use of a threshold weight w^*. In its weighted version, the CPM algorithm considers the existence of an edge $e_{i,j}$ between two vertices i, j if and only if the weight $w_{i,j}^{(d)} > w^*$. In order to determine the threshold weight w^*, Palla *et al.* propose to choose a value such that "the largest community becomes twice as big as the second largest one"[126], which is below the critical value w_{crit}^* for which a giant connected component arises [42].

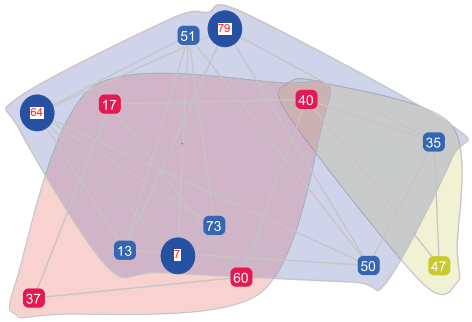
In our experiment, we calibrated the $\{w_q^*\}_{q=1}^T$ threshold values on a per-day basis, instead of keeping the same w^* throughout the trial. Because most of the participants followed one specific class that took place on Wednesdays, and the remaining days they might or might not have followed any common classes, we registered high RSSI proximity values on course days and more sparse values on non-course days. Hence, the per-day threshold $\{w_q^*\}_{q=1}^T$ was better suited for such bi-modal proximity patterns.

To illustrate the output of CPM, Figure 4.5(a)[2] and 4.5(b) show an example of the detected communities on a given day, based on the internal data and the observations of the adversary respectively. As it can be seen, some communities detected by the adversary are not present in the internal case; there is however an overlap between the members (the larger vertices) of the two sets of communities. We discuss and quantify this difference in Section 4.4.

[2]The figure is obtained by using the CFinder application developed by the authors of the CPM algorithm, freely available on www.cfinder.org.



(a) Communities inferred by using internal data.



(b) Reconstructed communities by the adversary.

Figure 4.5: Detected communities on a day T_k based on internal (local device) and external (adversarial estimates) data, respectively. The larger vertices are present in both community sets.

Community Statistics

In addition to detecting communities and their members, we compute five privacy-relevant and common community statistics $\{S_{(i)}(T_k)\}_{i=1}^{5}$ that will be compared in the accuracy evaluation process. In particular, for each day T_k of the trial we compute and compare the following statistics: S_1 is the community degree (the number of edges shared between two communities), S_2 is the distribution of the community size (the number of members of each community), S_3 is the community density (proportion of edges out of all possible edges relative to the sparsest set with $|C_i| - 1$ vertices), S_4 is the ratio of total out- and in-degree of communities and S_5 is the community membership value (the number of communities a vertex belongs to). The difference between the results obtained using the internal and external input data is defined by Eq. (4.2) as the ratio between

the absolute difference of the observed statistics over the maximum value

$$\Delta S_i(T_k) = \frac{|S_i^{ext}(T_k) - S_i^{int}(T_k)|}{\max\limits_{\forall T_k} \left(S_i^{ext}(T_k), S_i^{int}(T_k)\right)} \tag{4.2}$$

We have $\Delta S_i(T_k) = 0$ when the adversary's statistics is exactly the same as the statistics obtained using the internal proximity data, and $\Delta S_i(T_k) = 1$ when the two statistics have the largest discrepancy (or lowest similarity). We define the adversary's accuracy in inferring the community statistics as $1 - \Delta S_i(T_k)$.

Community Privacy

In addition to the differences in statistics $\Delta S_i(T_k)$, it is crucial to assess the similarity of the community composition in order to ascertain in a comprehensive way the privacy leakage of community information. To this end, we compute the well-established Jaccard index measure [89] for community similarity on each day T_k, which is a statistic that computes the similarity between two sample sets (or communities) C_i, C_j, where values close to zero mean that the adversary did not accurately infer the communities and their members, whereas values close to one indicate a very good adversarial accuracy in inferring the same communities. The Jaccard index is defined as

$$J(C_i, C_j, T_k) = \frac{|C_i(T_k) \bigcap C_j(T_k)|}{|C_i(T_k) \bigcup C_j(T_k)|} \tag{4.3}$$

In order to evaluate the adversary's accuracy of reconstruction of the communities in our pervasive network, we compute the Jaccard index on each day T_k between the communities $C_i(T_k)$, detected using internal device data, versus the reconstructed communities $C_j(T_k)$, detected using the adversarial estimated proximity information. Given $J(C_i, C_j, T_k)$ for each i, j on a day T_k, we define the Jaccard index matrix $JMat(T_k)$, where each element of the matrix is defined as $JMat(T_k)_{i,j} = J(C_i, C_j, T_k)$, i.e., the Jaccard index for all pairs of communities C_i and C_j. Without having access to the internal data, the adversary has no prior knowledge about which community C_i corresponds to which reconstructed community C_j. Therefore, in order to consider the best possible match for any pair of internal/reconstructed communities for each day T_k, we choose the match $(C_i(T_k), C_j(T_k))$ that maximizes $JMat(T_k)_{i,j}$. We then compute the aggregated Jaccard index over all such best matches as

$$JI(T_k) = \text{avg}_{\forall i} \left(\max_{\forall j} (JMat(T_k)_{i,j}) \right) \tag{4.4}$$

for each day T_k of the trial where there is at least one community detected by using both the internal and adversarial proximity information.

In the next section we quantify the community privacy leakage by computing the accuracy measure $1 - \Delta S_i(T_k)$, and similarity $JI(T_k)$ for each day T_k and weight function $\{w_{i,j}^{(d)}\}_{d=1}^{3}$, comparing the results obtained using the internal (local device) and external (adversarial) input data respectively.

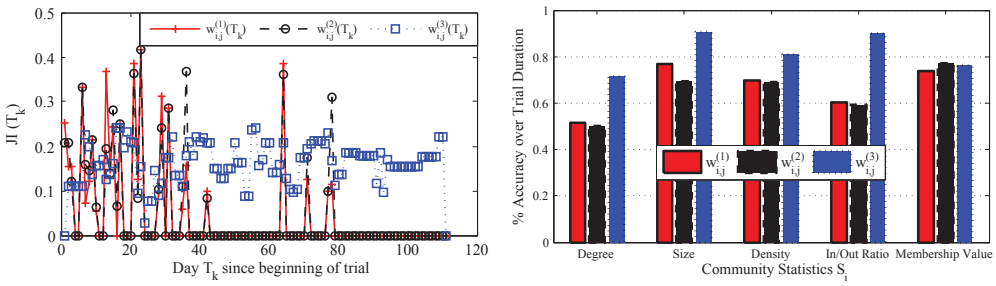
4.4 Results

In this section we provide the experimental evaluation of the privacy of pervasive communities through a comparative analysis of the adversary's accuracy of reconstruction of both community statistics and memberships. First, we evaluate the privacy across the three weight functions $\{w_{i,j}^{(d)}\}_{d=1}^{3}$ (inter-weight accuracy), by comparing the similarity between communities and the accuracy of their statistics obtained by using the internal (local device) proximity information with the external (adversarial estimates) data collected by the set of wireless sniffing stations. This will allow us to observe the evolution of the accuracy while increasing the sophistication of the weight functions, taking progressively into account several features of human and social behavior such as proximity, intensity, aging and recency of social relationships. Second, we perform an intra-weight comparison for the more realistic weight function $w_{i,j}^{(3)}$, in order to characterize the effect of the aging factor τ on the similarity and accuracy of community reconstruction attained by the adversary.

Figure 4.6 and 4.7 show the adversarial reconstruction similarity and accuracy results with respect to the communities detected using internal data, for the inter-weight and intra-weight scenarios respectively. For Figure 4.6(a) and 4.7(a), a value of $JI(T_k) = 0$ means that on day T_k there were no communities detected either using the internal proximity data or the external one. The complete list of the experimental parameters − selected in order to provide as much information as possible − can be found in the Appendix, which is provided as a supporting file to this document.

4.4.1 Inter-Weight Accuracy

By observing Figure 4.6(a), we first notice that the adversary is able to correctly reconstruct communities and identify their members in $20\% - 40\%$ of the cases, compared to the communities detected by using internal proximity data. In general, we observe that there is a significant difference in terms of similarity results between the first two weight functions $w_{i,j}^{(1)}, w_{i,j}^{(2)}$ and the third function $w_{i,j}^{(3)}$. The former two functions are solely based on the observations made on each particular day and independently of what happened in the previous days. Therefore one noticeable characteristic is the increased fluctuations in the similarity from one day to the other, which is a much less visible aspect for the latter weight function. As $w_{i,j}^{(1)}, w_{i,j}^{(2)}$ are very exposed to the periodicity of the course



(a) Community memberships similarity. (b) Community statistics accuracy.

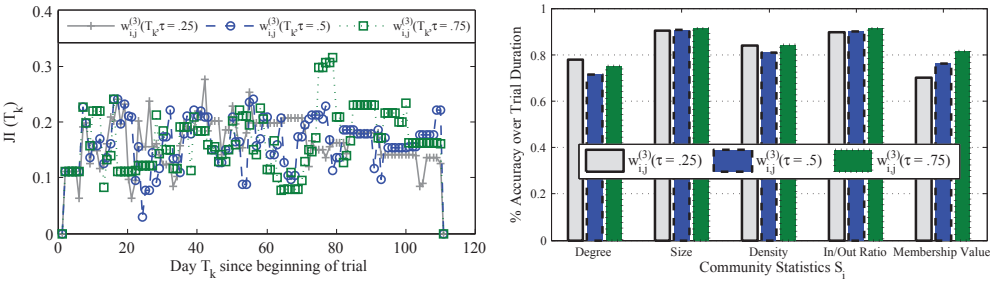
Figure 4.6: Adversary's accuracy of reconstruction of the pervasive communities for the three weight functions.

schedule of the participants, the adversary's similarity of reconstruction of the actual communities and their members greatly depends on the amount of data collected by his wireless mesh network. We notice that for the days when most students attended a particular class, the reconstruction similarity is higher (up to 40%) than for days in which students do not attend classes together. Hence even the two basic weight functions are able to provide a sensible similarity to the adversary when the users' movements are tracked by several sniffing stations simultaneously.

Contrary to $w_{i,j}^{(1)}$ and $w_{i,j}^{(2)}$, $w_{i,j}^{(3)}$ is able to capture more proximity information and allow the CPM algorithm to detect communities on the days in which the other two weight functions were unable to provide a sufficient amount of data. At the same time, however, the peaks of similarity tend to be much lower (25%) compared to the other functions. This suggests that $w_{i,j}^{(3)}$, while being able to produce more community information with scarce data, performs worse in the identification of the members in each community.

Regarding the difference in community statistics, depicted in Figure 4.6(b), we observe a better accuracy for $w_{i,j}^{(3)}$ compared to $w_{i,j}^{(1)}$ and $w_{i,j}^{(2)}$. In four out of five community statistics, $w_{i,j}^{(3)}$ has an almost 40% better accuracy compared to the other functions, which indicates that the former function provides better results on a higher structural community level rather than on an lower, individual community member level.

In general, we observe that all three weight functions are better able to produce accurate community statistics (Figure 4.6(b)) than to identify the correct community members (Figure 4.6(a)). In particular, $w_{i,j}^{(3)}$ shows that it is possible to achieve very accurate community statistics only by relying on externally collected data, thus shrinking the discrepancy between the community statistics based on internal data and adversarial's estimates down to 9%. This result indicates that, by collecting and analyzing radio information passively and without access to the devices themselves, an adversary is able to breach the privacy of



(a) Community memberships similarity.

(b) Community statistics accuracy.

Figure 4.7: Adversary's accuracy of reconstruction of the pervasive communities for three different values of the aging factor τ.

community information very successfully, although the more fine-grained identification of members of any given community remains a more challenging task.

4.4.2 Intra-Weight Accuracy

Figure 4.7(a) shows the adversary's performance in correctly identifying the communities and their individual members when using $w_{i,j}^{(3)}$ with three different values of the aging factor $\tau = \{.25, .5, .75\}$. According to its definition in Eq. (4.3.2), we assign an increasing coefficient to the past accumulated weight information $w_{i,j}^{(3)}(T_{k-1})$ in the computation of the current day's weight function $w_{i,j}^{(3)}(T_k)$. The goal is to study the effect of the "retention" of the intensity from the past on the privacy (or lack thereof) of community information.

One recurring characteristic, present also in the inter-weight comparison, is that the CPM algorithm detects communities in all days of the trial, independently of the amount of information available to the adversary on each particular day, even for a small value of τ. When $\tau = .25$, as expected the similarity fluctuates more when compared to $\tau = .5$, especially at the beginning of the trial.

However, Figure 4.7(a) shows that the stabilization of the similarity is not achieved by simply increasing the value of τ from .25 to .75; in fact, for the intermediate value of $\tau = .5$, we notice that the fluctuations are less pronounced than for a smaller or larger value. This suggests that, for relatively small or large values of the aging factor, the similarity achieved by the adversary tends to diverge more frequently from steady values, indicating that a stable value for the aging factor is more likely to be in the middle of the possible values [0.25,0.75], rather than at any of the extremes. When $\tau = .75$, the adversarial similarity increases sharply as the time passes, especially towards the end of the trial. This is somewhat surprising, as we would expect that by increasing the emphasis on the past − rather than on the current weight information − the similarity would be more stable when going through the trial. This is an interesting aspect to

consider in further studies on our community data.

When observing the results on the accuracy of the community statistics, as shown in Figure 4.7(b), we notice that, among the three considered values of τ, $\tau = .5$ is the least accurate, compared to smaller or larger values of τ. Moreover, in four out of five statistics, the largest value of $\tau = .75$ produces the best accuracy on average over the trial duration. This suggests that, although not converging towards a stable interval for the accuracy in identifying the communities and their members, putting more emphasis on the past accumulated information does increase (on average) the adversary's accuracy in computing correct community statistics using only passively collected data from fixed WiFi access points.

Overall, the results indicate that although less stable and more accurate at inferring community structures, emphasizing the past yields better accuracy for both community detection, identification of their members and for generic community statistics. This finding in particular is concerning in regard to privacy, as the amount of individual and community data that is collected by external parties might provide very accurate statistics, especially for group and community-targeted services. These results are significant, as they show how the message source ID, contained in almost any kind of radio message, not only is enough to provide accurate social community statistics, but it is also sufficient to successfully infer almost half of the members of such communities.

4.5 Related Work

The structural properties of short-lived communities in pervasive networks have been recently investigated from the performance [85, 31] and routing [83, 86, 31] perspectives; the authors of [38, 48, 72] investigated similar issues on the socio-behavioral level while studying people's preferences and group formation characteristics. For instance, it is shown that performance of packet-forwarding algorithms could greatly benefit from the human mobility and sporadic nature of inter-contacts [85], as the different connection frequencies between members of the same community with respect to members of other communities could significantly improve intra-community packet-forwarding while not disrupting inter-community communications. Similarly, [86] shows how forwarding performances similar to state-of-the-art algorithms could be achieved at a sensibly lower resource utilization if structural properties of communities are considered.

With respect to privacy, several works on location privacy address the risk and propose protection mechanisms for users' locations [13, 82, 90]. These contributions focus mostly on individual mobile users and their current neighbors. However, to the best of our knowledge, there is no prior study on the increasingly important issue of pervasive community privacy and its evaluation on a deployed network. This work constitutes the first building block for analysing community

privacy issues in pervasive networks.

4.6 Summary

In this chapter, we have addressed the important aspect of community privacy in pervasive networks. We conducted an experimental analysis of the adversary's accuracy of reconstruction of the communities. In particular, we focused on identifying the individual community members and on inferring the generic community statistics that are less dependant on the correct identification of individual users inside such communities.

Through a fine-grained characterization of the intensity of social contacts among people, we quantified the accuracy in both community reconstruction and community statistics for the whole duration of the trial, showing that even basic social intensity functions capture very accurately the generic statistics, such as the degree of a community, its size and density of links. However, reconstructing more specific information about the composition of each community and their individual members remains more challenging, even when using a more comprehensive model for characterizing the intensity of social relationships, which considers recency, ageing, and contact frequency in addition to proximity and duration. As a result, there is a substantial risk that accurate community information could be easily collected, inferred and misused by external third-parties, much to the detriment of users' community privacy.

Our results provide empirical evidence about the two distinct levels of community information leakage to external observers, who could be able to infer with high accuracy the different social groups and generic communities of people in pervasive networks, while being much less accurate in determining the affiliation of any particular individual to a community.

Publication: [22]

Chapter 5

Privacy of Social Relationships in Pervasive Networks

WiFi base stations are increasingly deployed in both public spaces and private companies, and the increase in their density poses a significant threat to the privacy of connected users. Prior studies, including the one presented in Chapter 4, provide evidence that it is possible to infer the social relationships and communities of users from their location and co-location traces, but they lack one important component: the comparison of the inference accuracy between an internal attacker (e.g., a curious application running on a mobile device) and a realistic external eavesdropper in the same field trial. In this chapter, we experimentally show that, for some social relationships, such an eavesdropper is able to infer the type of social relationships between mobile users better than an internal attacker. Moreover, our results indicate that by exploiting the underlying social community structure of mobile users, the reliability of the inference attacks increases by a factor of three. Based on our findings, we propose countermeasures to help users protect their privacy against eavesdroppers.

Chapter Outline In Section 5.1, we introduce the social relationship inference problem and present our goals. In Section 5.2, we introduce the experimental setup and system architecture. In Section 5.3, we detail the characterization of the social interactions and communities. We describe our relationship inference framework in Section 5.4 and its application to the experimental data. In Section 5.5, we present the results of the inference framework and their implications on the privacy of users' social relationships. We also discuss the limitations and perspective of this work, as well as countermeasures to help users protecting their privacy. We present the related work in Section 5.6 and we summarize the chapter in Section 5.7.

5.1 Introduction

Innovative Internet mobile services and products, such as location-based services and information-sharing platforms, enable users to enjoy a multitude of applications to easily stay in touch, work, have fun and exchange data. Beside online services accessed through infrastructure-dependent communications (e.g., WiFi hotspots and cell towers), upcoming near-field [116] and peer-to-peer (P2P) technologies, such as Nokia Instant Community (NIC)[36] and Qualcomm's AllJoin [131], allow for direct device-to-device connections, thus creating a new form of localized and context-aware interactions. For instance, Apple's AirDrop enables users to exchange files between mobile devices by creating an ad-hoc WiFi network. All in all, the amount of data exchanged wirelessly by mobile devices, be it in an infrastructure or P2P ad-hoc mode, has consistently increased over the past few years.

The information that can be inferred only by observing the data exchange patterns and users' (co-)location traces is of great importance and very sensitive: Home/work locations [81], activities [107], and social networks [38] are often data that people would prefer not to reveal publicly. Users' social relationships are no exception either, as several studies on ubiquitous computing have shown that location and co-location traces alone can reveal the presence and type of social relationships among people [47, 48]. Most of the existing works that evaluate the effectiveness of the inference of social relationships from location and proximity data are based on a single source of information: either proximity logs stored on the mobile devices or WiFi/cellular connection data obtained from the network operators. Although the existing results are significant and necessary for an initial attempt at tackling the issue of social relationship inference, they lack an important characteristic: data-source diversity for the same experimental settings. In other words, they lack the simultaneous access to proximity data present on both the mobile devices and in the operators' logs. Having access to these two data sets is paramount, as this enables us to evaluate and compare the accuracy in inferring the social relationships between an internal attacker, who has access to the data on the device, and an external eavesdropper who does not.

In this chapter, we address the problem of social relationships inference by carrying a comparative analysis of the exposure of such relationships and encounters in a deployed, peer-to-peer and infrastructure-based wireless network. We gathered communication and proximity information from 80 participants carrying smartphones during a four-month experiment. We collected complementary encounter information both from the smartphones and from the WiFi sniffing network, comprised of a set of access points (APs) passively intercepting communications between the participants; thus our study is the first to possess data-source diversity for the same experimental setting. These two sources reflect accurately the information that can be gathered by (i) a curious application running on a

smartphone or an experimenter and (ii) a network infrastructure owner such as a company which seeks to infer the social relationships between its employees or a network of compromised (community) access points; both of them represent realistic and practical adversarial scenarios. By exploiting such complementary data, we design a statistical inference framework to evaluate the accuracy of the inference of social relationships in localized and pervasive networks. This evaluation provides empirical evidence about the difference between an adversary's accuracy and that of a curious application in determining the type of social relationships between people. More specifically, our contribution is two-fold:

- We provide the first analysis, to the best of our knowledge, of the exposure of social relationships, based on proximity data, in a deployed adversarial WiFi network with data-source diversity.
- We design a statistical inference framework for relationship classification and use it to evaluate the success of the owner of the WiFi sniffing stations in inferring the type of social relationships between users and we then compare it against a curious application.

Our experimental results show that the infrastructure operator is able to infer some types of relationships between users better than a curious application that uses on-device proximity data. This is partially due to the fact that the external adversary has location information about the encounters between users. In addition, when exploiting the social community structure, the inference accuracy of the attacker is significantly higher. To put our results in perspective, we compare the strength of the considered adversary, in terms of the density of sniffing stations, to some existing hotspot networks.

5.2 System Architecture

In this section, we describe the methodology and experimental setup of our field experiment, along with the network model and the adversarial model.

5.2.1 Experiment Setup

The analysis presented in this chapter is based on the same experimental data collected in the summer 2011 on the EPFL campus. The detailed description of the experimental settings and the network model are can be found in Chapter 4 (Section 4.2).

5.2.2 Adversarial Model

The adversary considered in this chapter makes use of the same infrastructure presented in Chapter 4. However, the goal of the adversary in this chapter is

different. In order to infer the social relationships, the adversary collected the 3-tuple (Timestamp, Source MAC, RSSI) from the sniffed NIC messages. Note that at no time did the participants connect to the Internet through the adversarial APs, that were used only for passively sniffing the NIC packets. Due to the localized nature of the NIC message exchanges, they can be captured only by a local adversary. Such data arguably leaks an amount of personal information lower than the data that flows through the infrastructure. Because application-layer encryption was sometimes used to protect the message confidentiality, we assume that the adversary did not have access to such data. All adversarial knowledge is built from the analysis of the NIC data, passively collected by the sniffing APs.

In summary, the adversary performed the following: (1) Captured the NIC message exchanged by the participants' devices; (2) Built a social graph representing the presence and intensity of the encounters among users, based on the estimated physical proximity between their mobile devices; (3) Inferred the social communities of the participants, based on the social graph and the intensities of their pairwise interactions; (4) Inferred the type of social relationships between participants by using our inference framework (presented in Section 5.4, the localized communities and the trained relationship inference framework. Note that the adversary usually has some background knowledge about the types of social relationships to be inferred, based on the targeted population and context (in our case, students on a campus).

5.3 Social Interactions and Communities

In this section, we describe the methodology used to model the user interactions during the experiment. The social graph representing the user interaction is used to infer the community structure, which is subsequently used to refine the social relationships inference mechanism, as explained in Section 5.5.

Following the notation for social communities defined in Chapter 4, we represent people and their relationships by an undirected weighted graph $G{=}(V, E, W)$, where the vertex set V corresponds to people, the edge set E expresses the existence of a relationship between people, and the weight function $W : E \rightarrow \mathbb{R}^+$ quantifies the intensity of such relationships. In their simplest form, communities can be represented as subgraphs $\{C_i = (V_i, E_i, W)\}_{i=1}^{M}$, where $C_i \subseteq G$ and M is the number of communities. As people are usually members of several communities, different community subgraphs can share vertices.

5.3.1 Experimental Data

We model the collected encounter data from the experiment as a social graph G. Hereafter, we describe the type of information that is used to define the existence

and intensity of relationships between users. In our experimental data, we have two sources of proximity information: (i) the *internal data*, constituted by local device logs collected by the mobile devices themselves and containing encounter data (list of neighbors' MAC addresses, the timestamps and the RSSI values of received packets), and (ii) the *adversarial data*, containing the headers of the sniffed packets (sent by the mobile devices), which include the timestamps and RSSI values of received packets at the sniffing APs, as well as the device ID of the sender.

We use these two data sources to formulate the 'strength' or intensity of the social relationships between users and to define the weights of the edges that connect the respective vertices in the social graph $G = (V, E, W)$. From the local device logs, we can directly obtain the device-to-device proximity information because the recorded RSSI values on the receiving devices are correlated with the real distances to the sending devices. However, this is not the case for the RSSI values recorded by the adversarial network, as they depend on the distance between the sending device and the receiving sniffing station. Following a similar approach as in Chapter 4, we estimate the position of a mobile device that sent a packet based on the RSSI at all sniffing stations that overheard the packet.

In our social graph, we consider three distinct weight functions $\{w_{i,j}^{(d)}\}_{d=1}^{3}$ for the edges between vertices $i, j \in V$ to quantify the intensity of their social interactions. The three weight functions [22] progressively take into account the proximity, the intensity, and the aging/recency of the relationships between users. Moreover, to investigate the effect of community detection on the accuracy of the inference attack, we apply the well-suited CPM method[126] to extract community information from the social graph G.

5.3.2 Relationship Labels

In addition to the strength of social relationships, we also characterize the type of relationship between users. More specifically, we consider three labels (i) *friends*, (ii) *classmates* and (iii) *others*. The possibility of associating multiple labels to a single relationship is crucial in social networks, as people can be simultaneously classmates and friends. To construct the ground truth data, each relationship is assigned one (or more) labels based on (i) the participants' answers to the survey questionnaires (for friends) and (ii) the database of academic course registration provided by the university (for classmates). For instance, if two participants followed the same class during the experiment, their relationship was categorized as *classmates*; furthermore, if one of the same participants also listed the other as *friends* in the questionnaires, their relationship was categorized as both *classmates* and *friends*. We note here that for the friendship label, due to the lower number of declared friendships, we considered asymmetric relationships as symmetric because of the subjective nature of the friendship relationship that

could be perceived by one person and not by the other. From our ground truth data, we observed that 25% of all the 3,160 possible relationships were tagged as *classmates* and 2% as *friends*. We tagged the remaining as *others*.

We choose these types of relationships for two reasons. First, we followed the approach of [25] and selected the types of relationship relevant to our context. For instance, as our population was comprised almost exclusively by students on a university campus, we considered the two relationship types aforementioned, as well as the estimated user location. For a more diverse on-campus population, including researchers and instructors, we would have had to consider the role of the hierarchy as well. Second, it was possible for us to collect the ground truth for these two types of relationships.

5.4 Relationship Inference Framework

In this section, we introduce our relationship inference framework by which we evaluate the accuracy of the attacker's reconstruction of users' social relationships on our data sets. First, we describe the method we adopt for characterizing a relationship from the internal and adversarial (estimated) proximity data. Second, we describe each component of the framework and we explain how it exploits the two different data sets.

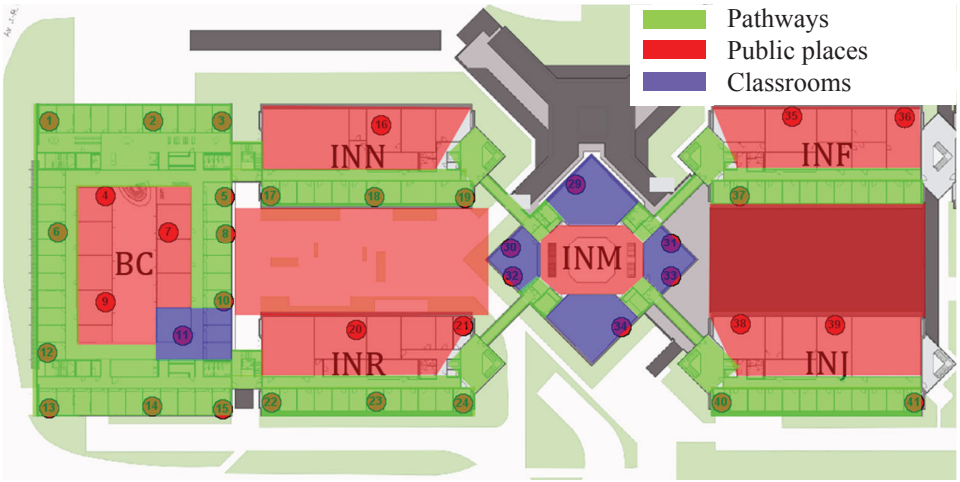


Figure 5.1: Partition of the map (in which the experiment was conducted) into three types of areas: pathways, public places, and classrooms.

5.4.1 Relationship Characterization

The relationships between two users are characterized by their interactions. To characterize and classify relationships, we first formalize the notion of *encounter*:

it captures a *significant* interaction between two users. A relationship is defined by a set of encounters.

Encounter Detection

We define an encounter between any two users as a continuous time interval that meets a threshold, both on its duration and on the proximity of the two involved users. Typical values of these thresholds are a few minutes and a few dozens of meters, respectively. The proximity between users is computed in two ways, according to the source of the data. From the internal device logs, we use the RSSI values of messages exchanged between users (in a P2P fashion) in order to determine their distance, whereas from the adversarial sniffed data, the proximity is determined by computing the Euclidian distance between the estimated location coordinates of each user, based on device-to-AP RSSI values.

A number of features are extracted from the internal/adversarial logs, either at the granularity of an encounter or of a relationship. For instance, we extract the duration of each single encounter and the total number of encounters between the two users for each relationship. The extracted features are then fed to the classification algorithm. In order for the adversary to accurately classify the relationships based on specific features, these features must vary significantly from a category (such as *friends*) to another. Typical discriminating features include encounter duration, the proximity of users during encounters, inter-encounter time, and the number and periodicity of encounters [159].

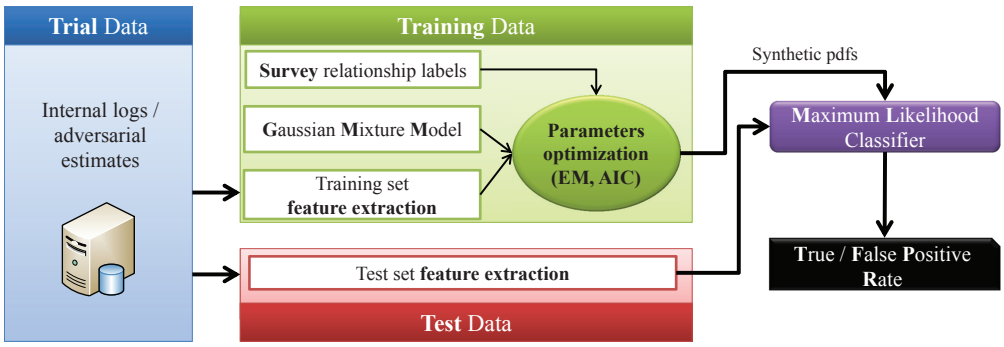


Figure 5.2: Overview of the inference framework.

5.4.2 Relationship Classification

In order to select the set of possible relationship classes, we use the ground-truth collected through (i) survey questionnaires at the beginning of the trial and (ii) the official university course-registration database. We then designed an inference framework (Figure 5.2) based on the features extracted from the internal/adversarial data.

The classification algorithm operates as follows. First, we extract the features of a small subset of the encounters and relationships, namely a training set. From the survey and administrative records, we know the category of each encounter and relationship in the training set. Then, for each of the three categories of relationships, we compute the experimental *probability density function* (pdf) of the various features and plot the corresponding histograms. We fit each experimental probability density function with an appropriate parametric model, the Gaussian Mixture Model (GMM) [47], and determine the parameters according to two optimization criteria. The obtained synthetic pdfs are subsequently used to classify the encounters of the rest of the data set, i.e., the test set. Based on the synthetic pdfs, for each encounter and for each category of relationships, the likelihood that the encounter belongs to this category is computed from the values of its features. The likelihood that a relationship belongs to a given category is computed by taking into account the different features of the relationship and those of its encounters.

Encounter Location Classification

In our context, the location where encounters take place is a key feature of relationships. This information is only available in the adversarial data and is computed from the users' locations estimated by the adversary. More specifically, the region of the experiment covered by the adversarial network is partitioned into different areas classified in three types: *pathways*, *public spaces* and *classrooms* (Figure 5.1). We expect such a classification to be discriminating, with respect to the types of relationships the adversary wants to infer. For instance, classmates are expected to experience frequent encounters mostly in classrooms, whereas friends might hangout more frequently in public spaces and pathways. Because users can move during a given encounter, for each encounter, we compute the proportion of the time spent in each type of location. In this way, we can capture the fact that an encounter that started in a pathway continues through public spaces.

Feature Extraction

We detect encounters based on interaction duration and proximity between users. With the internal data, we assume that an encounter occurred if at least two messages are exchanged within an interval of 5 minutes and the average RSSI of the exchanged messages is greater than -80 dBm (to account for radio noise, interference and fading), which corresponds to a distance of at most 18 meters. These threshold values have been determined empirically. The encounter spans from the first exchanged message to the last. For example, if two users exchange messages, all with RSSI values of -50 dBm, at times 1, 3, 7, and 15 minutes, an encounter is detected and it spans from 1 to 7 minutes With the adversarial data,

we first need to estimate the users' locations. In order to reduce the noise on the users' estimated locations, we divide the trial timeline in small sub-intervals of 30 seconds and we average user locations over each sub-interval. First, we consider that an encounter occurs if at least two messages are exchanged within an interval of 5 minutes and the distance between the two users' locations is consistently lower than 5 meters (i.e., lower than 5 meters in each sub-interval between the times at which the messages were exchanged). A distance of 5 meters between two devices translates to an RSSI value of around -53 dBm. To compare the results with the internal data, we also considered distance thresholds of 10, 20, 30, and 40 meters, corresponding to RSSI values of -68, -82, -91, and -97 dBm, respectively.

At the granularity of an encounter, we extract the following features: (i) the **encounter duration** (the time elapsed between the first and last message exchanged during the encounter), (ii) the **inter-encounter time** (the time elapsed since the end of the previous encounter and the beginning of the current encounter), and (iii) the **average encounter RSSI** value (the average of the RSSI values of the messages exchanged during the encounter). For the adversarial data, we further consider (iv) the **encounter location**, characterized by the fraction of the encounter time that takes place at each type of location, such as pathways, public places and classrooms. The location of an encounter is defined as the midpoint of the two users. At the granularity of a relationship, we consider an additional feature: the total **number of encounters** over the whole trial.

Supervised Learning

In order to train our inference framework, we use a subset (the training set) of all the relationships, which account for approximately 30% of the whole data set. We divide the relationships in the training set into three categories, i.e., *friends*, *classmates*, and *others* according to the labels obtained from the surveys and the university database. Because two users can simultaneously be friends and classmates, a relationship can belong to both categories. Note that an alternative way to proceed would be to divide the training set into four categories: *friends*, *classmates*, *others*, and *friends and classmates*. However, as in our data set the latter category does not contain enough relationships to perform a proper training, we consider the former three categories. For each category, and for each feature, we compute the experimental distribution of the feature for all the relationships in the category, by means of histograms composed of 100 bins. We denote by $\tilde{f}^{\text{feat}}_{\text{cat}}$, cat $\in$ {friends (f), classmates (c), others (o)} and feat $\in$ {encounter duration (ed), inter-encounter time (iet), average RSSI (rssi), and number of encounters (ne)} (and encounter location (el) for the adversarial data), the functions corresponding to the histograms. For example, $\tilde{f}^{\text{ed}}_{\text{f}}(x)$ gives the proportion, in the training set, of encounters between friends that last between

x and $x + \delta x$ minutes. These functions can be thought of as indicators of the likelihood that a given encounter belongs to a given category, based on the value of one of its features, and will be used as such by our classification algorithm.

Intuitively, it can be expected that the pairs of users whose relationships are classified as *friends* and *classmates* would experience more frequent encounters than users belonging to the category *others*. Similarly, classmates are expected to meet according to a fixed pattern (e.g., every Monday for a particular class), whereas on-campus encounters between friends are not expected to follow a fixed pattern. This intuition is confirmed by our observation of the data, as shown in Figures 5.3 and 5.4. The former depicts the experimental distribution of two features: the encounter duration (at the granularity of an encounter) and the number encounters (at the granularity of a relationship), whereas the latter shows the location feature. It can be observed that, as expected, these features discriminate the three categories of relationships, therefore we foresee a high potential for classification. For instance, *friends* meet more than *classmates* and *others* and *classmates* meet for longer times (classes). Also, *classmates* meet mostly in classrooms. Note that the aforementioned intuitions are given for the sake of information, the inference algorithm is based on the rigorous techniques that we describe.

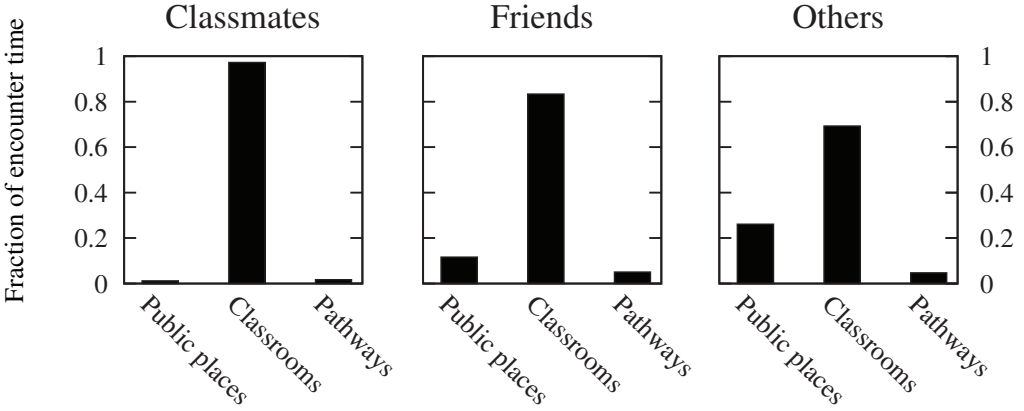


Figure 5.3: Encounter location feature for the adversarial data (distance threshold of 40 meters).

As it can be observed in Figure 5.4, the experimental distributions are noisy and thus cannot be used as is for classification purposes: For instance, we can observe in the histograms that there are no relationships between friends composed of exactly 35 encounters in our training set. Therefore, if the histogram of the number of encounters is used as is to determine the likelihood of two users being friends, two users who meet 35 times during the trial would be assigned a null likelihood for the *friends* category. For this reason, we fit the experimental histograms with parametric pdfs, namely multi-dimensional Gaussian mixture

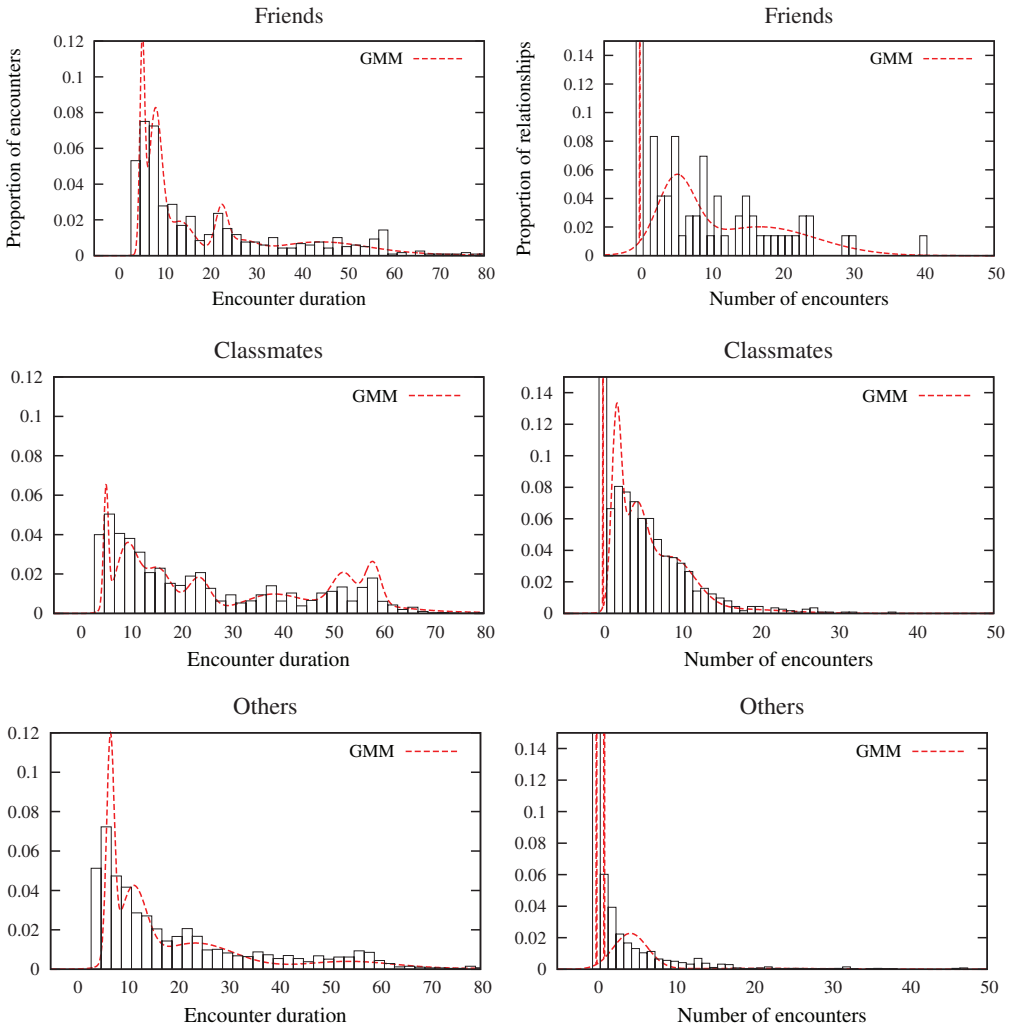


Figure 5.4: Experimental distributions (histograms) and fitted GMM pdfs (dashed lines) of two features extracted from the adversarial data (distance threshold of 40 meters): encounter duration and number encounters. From top to bottom, we show the GMM pdfs for the class labels *friends, classmates* and *others*, repectively.

models of the form:

$$p(\mathbf{x};\theta) = \sum_{k=1}^{K} \pi_k \frac{1}{\sqrt{|\Sigma_k|}(2\pi)^{d/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)\Sigma_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)},$$

where $\sum_k \pi_k = 1$ and d is the dimension of the feature vector. The set of parameters is denoted by $\theta = ((\pi_1, \boldsymbol{\mu}_1, \Sigma_1), \ldots, (\pi_K, \boldsymbol{\mu}_K, \Sigma_K))$, and $\mathbf{x}$ is the value of a feature. We use one-dimensional GMMs for scalar features, e.g, encounter duration, and multi-dimensional GMMs for the encounter location (fractions of time spent in pathways, classrooms, and public places).

The number K of Gaussian components, and their respective weights (π_k), means ($\boldsymbol{\mu}_k$), and covariance matrices (Σ_k) are free parameters and must be chosen to best fit the experimental distributions. For a given value of K, we determine the other free parameters of the Gaussian components through expectation maximization (EM). We choose the values of the parameters that maximize the likelihood that the n values $\{\mathbf{x}_i\}_{i=1..n}$ of the features (observed in the training set) have been drawn from a Gaussian mixture with these parameters. The likelihood function is given by

$$\ell(\theta; \mathbf{x}_1, \ldots \mathbf{x}_n) = \prod_{i=1}^{n} p(\mathbf{x}_i; \theta),$$

and the estimated value of the parameters is $\hat{\theta} = \arg\max_\theta \ \ell(\theta; \mathbf{x}_1, \ldots \mathbf{x}_n)$. We repeat this estimation step for several values of K, and we subsequently determine the most appropriate value of K according to the Akaike information criterion [1]: $\mathrm{AIC} = 2k' - 2\ln \ell(\theta)$, where k' is the number of parameters of the GMM.

The learning phase produces a likelihood function $\ell^{\mathrm{feat}}_{\mathrm{cat}}$ for each category $\mathrm{cat} \in \{f, c, o\}$ and for each feature $\mathrm{feat} \in \{\mathrm{ed}, \mathrm{iet}, \mathrm{rssi}, \mathrm{ne}\}$ (and el for the adversarial data).

Relationship Classifier

Our classification algorithm is based on likelihood maximization and operates in two steps. We first determine if the two users are actually involved in a relationship (friends or classmates) or not (others) based on the likelihood of a relationship, with respect to each category. This likelihood is computed from the value of the features and from the synthetic distribution obtained from the training set. If users are believed to have a relationship, we refine the classification by discriminating between friends and classmates.

More specifically, the algorithm operates as follows (see Figure 5.5 for illustration). If the likelihood is maximized for the *others* category, the relationship is classified as *others* and the classification ends. If it is maximized for the *friends* category, the relationship is assigned the *friends* label and a second step is performed to decide whether the *classmates* label is assigned as well. This second step is based on a threshold: If the likelihood of the relationship for the category *classmates* is higher than the likelihood of *friends* multiplied by a factor $\gamma \in [0, 1]$, the relationship is assigned the *classmates* label as well. The same process applies in the case the likelihood is maximized for the *classmates* category. The classification process can be summarized by the decision tree depicted in Figure 5.5.

The likelihood ℓ_{cat} of a relationship, with respect to each category cat, is given by

$$\ell_{\mathrm{cat}}(r) = \prod_{e \in r} \ell^{\mathrm{ed}}_{\mathrm{cat}}(\mathrm{ed(e)}) \cdot \ell^{\mathrm{iet}}_{\mathrm{cat}}(\mathrm{iet(e)}) \cdot \ell^{\mathrm{rssi}}_{\mathrm{cat}}(\mathrm{rssi(e)}) \times \ell_{\mathrm{cat}}(\mathrm{ne(r)}) \qquad (5.1)$$

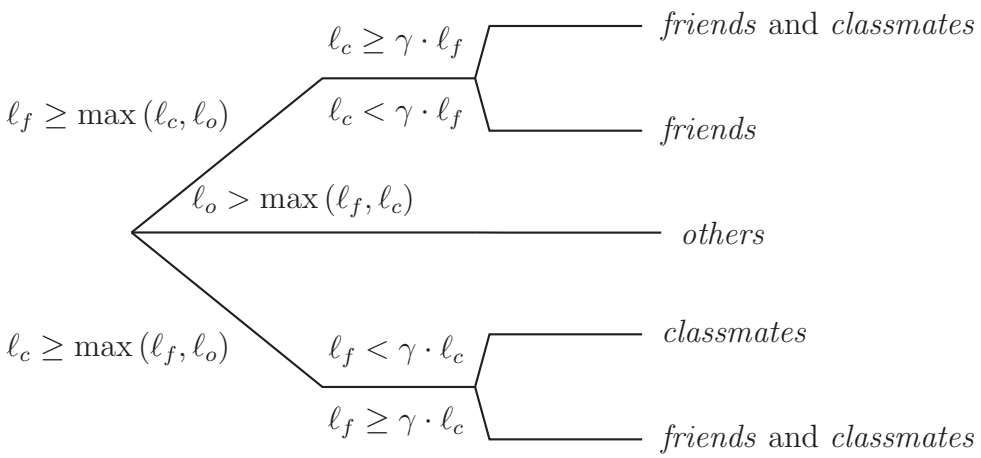


Figure 5.5: Decision tree used by the classifier.

where $m \in r$ denotes the encounters between the two users concerned by the relationship r. For the adversarial data, the factor $\ell_{\text{el}}(\text{el}(e))$ must be inserted in the product in Eq. (5.1) to take location into account.

5.5 Results

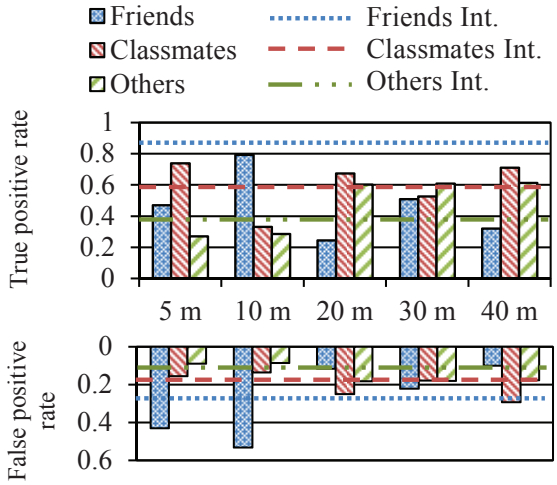
In this section, we present the performance results of our relationship classification inference framework. First, we describe the metrics used to evaluate the inference accuracy of the adversary, then we discuss the performance of the classifier and the effect on the users' relationship privacy. Finally, based on our results, we discuss possible countermeasures to mitigate the success of the adversary in inferring social relationships. We evaluate his success by using both pairwise-only and community-enhanced proximity information.

5.5.1 Pairwise Relationship Inference

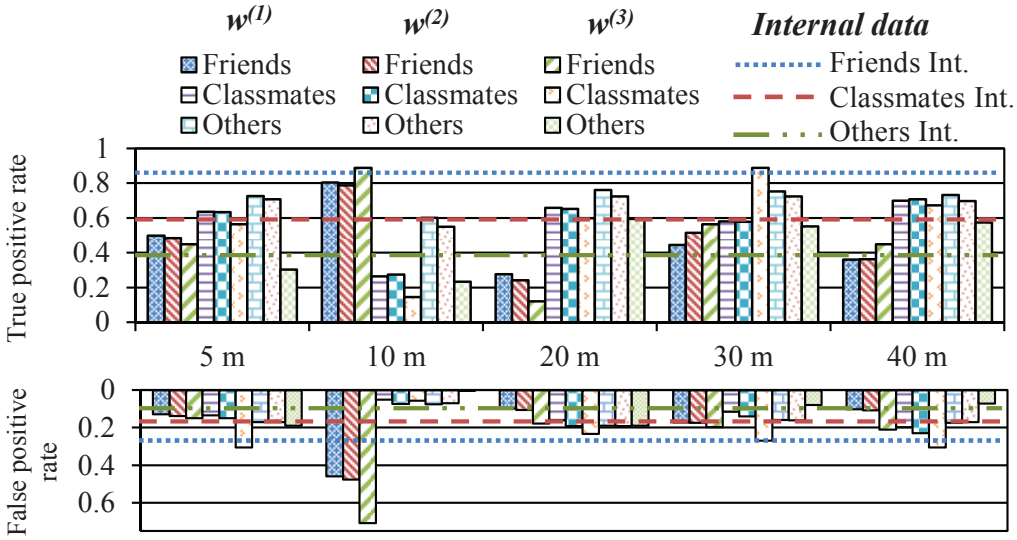
The performance of a classifier is usually evaluated in terms of its *true positive rate* (TPR) and *false positive rate* (FPR). Hereafter, we present the accuracy of the adversary's relationship inference on our data set, where the adversary considers only the estimated pairwise proximity information between users, i.e., without taking into account the underlying community structure. Figure 5.6(a) shows the TPR and FPR for the three classes for different distance thresholds.

Internal Proximity Data

By using the internal proximity information between users, the performance of our inference framework is, in general, satisfactory, given the limited amount of training samples (30% of the total). This means that we have a high TPR and a moderate FPR. For the class *friends*, we observe the best TPR (84%)



(a) Pairwise-based classifier accuracy.



(b) Community-enhanced classifier accuracy.

Figure 5.6: Accuracy of the external adversary in inferring social relationships. The horizontal lines correspond to the accuracy of the classifier when the internal (local device) proximity data is used.

and a moderate FPR (27%), which means that the actual proximity data and the encounter features we selected for the classification are quite discriminating for this class. Classmates and other relationship classes have a TPR/FPR of 56%/18% and 37%/13% respectively, which are lower than for friends. As a result, the actual proximity data works well for the friends and classmates relationships inference, whereas it has a more limited success in classifying other types of relationships. This is not surprising, as proximity information, without taking into account the actual location of the interaction, is intuitively tied to inferring

close and periodic interactions more than sporadic encounters.

Adversarial (Estimated) Proximity Data

In general, the adversary's accuracy in inferring users' social relationships has a bi-modal behavior with respect to TPR and FPR. First, at a lower encounter-distance threshold (5-10 meters), the adversary has a very high TPR rate for classmates and friends (79% and 74% respectively) while it suffers from an elevated FPR as well (53% for *friends*). The accuracy for *others* is relatively low at 28%, meaning that the prediction suffers from a significant number of false positives that negatively affect the successful recognition of friends and classmates. Whereas, for larger encounter-distance thresholds (20-40 meters), the accuracy for *others* doubles and is consistently larger than for the *friends*.

Internal vs. Adversarial

For low encounter-distance thresholds, on one hand the external adversary has a similar or higher accuracy in correctly inferring the relationships of participants compared to the case of the internal adversary (i.e., using the actual proximity data). On the other hand, the FPR is also substantially higher, which renders the relationship prediction much less reliable at low distance thresholds. As the adversary can estimate, in addition to users' proximity, some contextual information (location of the encounter), it is easier for him to infer the type of relationship given this additional feature, but it is also easier to wrongly include other types of relationships in the *friends* class.

When increasing the distance threshold (20-40 meters), we can see that the success in inferring friendships from external data decreases by at least 50% compared to the case where internal data is used. Meanwhile, there is a two-fold increase in TPR for the *others* category. In other words, the larger the allowed distance between users is, the better the inference accuracy for the *others* category. And the *classmates* performance stays at comparable levels for the different distance thresholds. Figure 5.7 shows the ROC surface (TPR vs FPR) of our statistical inference framework for different distance thresholds, considering pairwise-only (left) and community-enhanced (right) interactions.

Overall, we observe that the increase of the encounter-distance threshold for the adversary is highly beneficial for the detection of non-friendships. By adjusting such a threshold based on the target relationship class, the external adversary obtains an inference performance comparable to that of the internal adversary who uses the actual proximity information available on the devices.

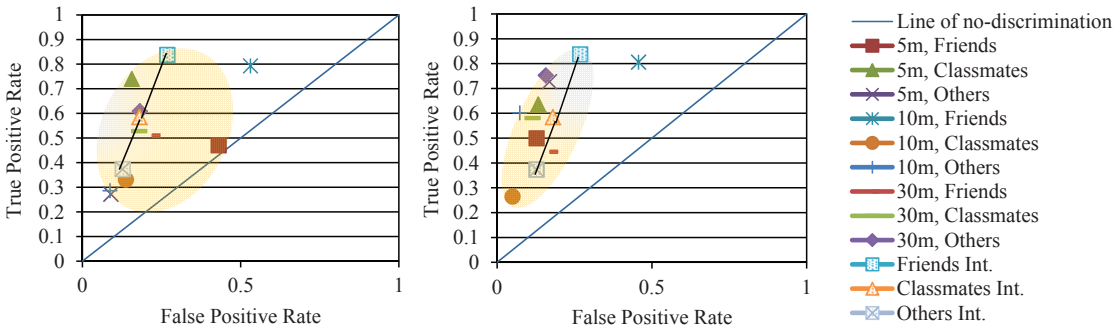


Figure 5.7: Receiver Operating Characteristic curve. On the left chart, the TPR/FPR values are based on pairwise-only proximity data. On the right chart, the TPR/FPR values are based on the community-enhanced proximity data.

5.5.2 Community-Enhanced Relationship Inference Results

Hereafter, we describe the performance of the relationship inference framework when the adversary relies on the underlying structure of user communities. This pre-processing step will enable us to observe the effect of such information on the accuracy of the adversary's inference compared to the previous case without community information. The results are shown in Figure 5.6(b).

Internal vs. Adversarial

For low-distance threshold values (5 meters), we see that the external adversary has a TPR comparable to the case where no community information is extracted (for most weight functions). However, the FPR values are significantly different: The false-positive rate for friends and classmates relationships is three to four times lower when using community structure than when not. When inferring the *others* class of relationships, the TPR is three times better than the community-less counterpart, with only a comparable FPR. These results indicate that by using the community structure of social relationships among users, the adversary is able to significantly increase his performance (both in terms of TPR and FPR) across all relationship types, especially in terms of false positives. With respect to the weight functions we defined for the CPM community detection algorithm, we can see that the first two functions $w^{(1)}$ and $w^{(2)}$ have a slightly better performance for friend and classmate relationships and are twice as good for the *others* class. This suggests that by modeling the interactions of non-friends or classmates with memory-less weight functions, the detected community structure is better suited for distinguishing between such types of relationship, as they did not exhibit most of the periodic and close interactions during our experiment as friends or classmates.

5.5.3 Discussion

The results we have presented so far show that, even without any additional pre-processing by the adversary, the estimated location and proximity among mobile users is often enough to make predictions about the kind of social relationship between any two users (Figure 5.6(a)). Even if the users rely on encryption for their communications, the considered external adversary can distinguish users as friends, classmates or neither by capturing their identifiers and by correlating them over time and space.

Even more striking is the fact that by adding an extra step in the inference process, both the accuracy (TPR) and reliability (FPR) are improved significantly. In particular, the dynamic social community structure of people and their behavior over time leak a substantial amount of additional private information to the adversary. By pre-processing the proximity data and detecting communities, in most cases the adversary is able to double his performance when inferring the types of relationships between users inside communities, much to the detriment of the privacy of users and their communities. This suggests that even a simplistic characterization of human encounters, based on memory-less and time-independent functions, GPS-less data and standard tri-lateration positioning algorithms, is enough to enable a wireless infrastructure owner to successfully detect communities and to determine the type of relationships among users inside each community.

Finally, by transposing our results to existing wireless networks, we hypothesize that similar findings can be obtained by self-interested or curious third-parties, such as commercial service providers and cellular network operators. The increasing deployment of low-range cellular base stations [54] and the increased availability of public WiFi hotspots [56] severely undermine people's ability to resist or opt-out from unwanted scrutiny.

5.5.4 Countermeasures and Limitations

Countermeasures

A solution for limiting the success of the inference attack, which is specific to the threat presented in this chapter, consists in reducing the information available to the adversary and his confidence in this information. To achieve this, technical solutions include the use of changing pseudonyms as wireless identifiers and sending dummy messages. With pseudonyms, users can change their MAC address every day while off-campus or, upon encounters, collectively interrupt all communications for a certain time and reconnect with fresh random pseudonyms (i.e., mix-zones [13]). The use of pseudonyms makes it difficult for the adversary to link and track identities of users over time. It reduces the learning period to a few hours instead of a the whole trial timespan, therefore decreasing the success of

the inference. However, such a mechanism has a negative effect on the usability of the communication features, e.g., Bluetooth pairing and MAC filtering. Another technical countermeasure is to dynamically change the transmission power to confuse RSSI-based triangulation and limit the eavesdropping possibilities. However this comes at the cost of decreased networking performances. Finally, injecting dummy messages would artificially increase the intensity of the encounters, thus, biasing the results of the inference.

Limitations

The results and discussions presented in this study are based on a large-scale experiment conducted on a university campus, where most of the participants are students. The characteristics of such a population are rather homogeneous, with respect to the utilization of technology and the age. As in several other university-based field experiments, the results of our study might apply – to a lower extent – to scenarios involving a more diverse population with respect to education, age, gender and technology usage habits. In order to mitigate this, as part of our future work we would like to extend our experiment to a more diverse population, and therefore to assess the effectiveness of the inference attacks in a more mixed environment. Finally, as off-campus interactions are more insightful with respect to social relationships – for instance, the larger variety of location types (e.g., bars, cinemas and residential areas instead of just public places and classrooms) – we expect a greatly improved performance of the inference when considering off-campus interactions as well. This would enable us to consider fine-grained social relationships, such as close friends and relatives.

Strength of the Adversary Regarding the relative strength of the adversary considered in the chapter, in terms of the number of APs, we studied the density of three deployed WiFi AP networks for comparison. We looked at the FreeWifi [60] and SFRWifi [141] hotspots networks that consist of the wireless modems and set-top boxes of the subscribers of two major ISPs operating in France. Such networks of hotspots constitute concrete examples of an adversary as they are operated by a single entity. For the Paris area, the average density is 564±270 APs/km^2 (resp. 853±346 APs/km^2), and goes up to 1450 APs/km^2 (resp. 1560 APs/km^2) in populated areas for Free (resp. SFR). For comparison, the adversarial network of sniffing APs we considered has a density of 1138 APs/km^2 (37 APs deployed over a region of size 130 m×250 m). Another example is that of a company exploiting its WiFi stations to infer the social relationships between its employees. For instance, our university has ∼880 WiFi APs deployed over a region of 500 m×800 m, that is a density of 2200 APs/km^2. These statistics show that our results are realistic and applicable to existing networks.

5.6 Related Work

Eagle et al. [48, 47] studied complex social systems by collecting Bluetooth encounter data from 94 students equipped with mobile phones. In particular, they identified proximity patterns between devices and performed relationship inference on the data by comparing the results with the ground truth obtained from surveys and answers to questionnaires. Similarly to other user studies [43, 117], the authors could only compare their findings (based on Bluetooth encounters between devices) with the ground truth, without having access to a network infrastructure that would allow them a more thorough analysis by comparing these results with those obtained by a passive third-party adversary. However, off-campus location data was collected in [48, 47] and it significantly improved the quality of the inference results. A recent study [9], performed on a group of 27 participants, shows that it is possible to accurately infer the social relationships between people based on Bluetooth encounters, phone calls, SMSs and cell-tower IDs, without any prior knowledge about the participants. Nevertheless, these findings are based on data that is available only on the local devices, therefore it is unclear to what extent the results apply to the inference success of an external adversary. Other studies focused on inferring the social relationships from co-presence of users, as in the case of Flickr [38], and on dynamics of inter-encounter times [159]. A large user study, involving 168 participants over 2 years, was conducted in [94] in order to study people's behavior with respect to mobile phone usage, activity and location.With regard to location privacy in pervasive wireless networks, [23] studied the efficacy of dynamic allocation of pseudo-random IDs in specific regions called *mix-zones* [13], showing that in deployed networks such a mechanism provides only limited success in protecting users' location privacy.

By relying on the graph-theoretic analysis introduced and evaluated in [22], our work uses weight functions and community structures that are well suited for the analysis and inference of social relationships in a pervasive wireless network. In contrast to the former work, in this chapter we focus on the accuracy of the adversary in classifying the type of social relationships among users, and we study the effect of using the community structure on the performance of the classification.

5.7 Summary

In this work, we have experimentally evaluated a practical adversary's accuracy in inferring the type of relationship of users in ubiquitous networks. The availability of source-diversity in our experiment enables us to compare the accuracy of an external adversary with that of an internal adversary (e.g., a malicious application) who has access to actual proximity information stored on the devices. Our results show the following two aspects. First, social relationships are exposed

to a significant inference threat by an external adversary, who can quite accurately infer the type of social relationships (friend, classmate and other) between users simply by relying on location estimates and encounter characteristics, compared to a curious application running on the device (or to the experimenter). Second, by applying a well-suited community detection algorithm, the adversary can double his inference accuracy on the same data set.

Publication: [17]

Part III

Cross-Layer Approach for Privacy and Information Sharing

Chapter 6

Adaptive Information-Sharing in Mobile Social Networks

Personal and contextual information is increasingly shared via mobile social networks. Users' locations, activities and their co-presence can be shared easily with online "friends", as their smartphones already access such information from embedded sensors and storage. Yet, people usually exhibit selective sharing behavior depending on contextual attributes, thus showing that privacy, utility, and usability are paramount to the success of such online services. In this chapter, we present SPISM, a novel information-sharing system that decides (semi-)automatically whether to share information with others and at what granularity, whenever they request it. Based on active machine learning and context, SPISM adapts to each user's behavior and it predicts the level of detail for each sharing decision, without revealing any personal information to a third-party. Based on a personalized survey about information sharing involving 70 participants, our results provide insight into the most influential features behind a sharing decision. Moreover, we investigate the reasons for the users' decisions and their confidence in them. We show that SPISM outperforms other kinds of global and individual policies, by achieving up to 90% of correct decisions.

Chapter Outline In Section 6.1, we present the topic and formulate the goals of this chapter. In Section 6.2, we introduce the SPISM information-sharing platform, including the system architecture, the operating principles and the decision-making core. In Section 6.3, we present our user-study and the methodology. We then evaluate SPISM and present the results from the study in Section 6.4. We discuss the related work in Section 6.5, and we summarize the chapter in Section 6.6.

6.1 Introduction

Mobile social networks are becoming extremely popular. Every month, more than 250 million people use their smartphones in order to get the latest updates from their favorite social networks[1]. Having access to users' personal data and physical context (through an increasing number of embedded sensors), mobile devices represent a simple means to quickly share information with others; location and photos are just two examples of data that can be easily shared. In addition to the user-triggered sharing decisions, applications such as FourSquare and the now-closed Gowalla enable users to configure their smartphones to share their location and co-presence automatically. With a small set of default information-sharing policies, users have the possibility to adjust the settings in order to match their sharing behaviors with their privacy concerns.

Prior studies on sharing behavior in mobile social networks have investigated the issues related to contextual information-sharing [144, 152, 12, 136]. By analyzing people's sharing behaviors in different contexts, they show that it is possible to determine the features that most influence users' sharing decisions, such as the identity of the person that is requesting the information and the current location[152]. For instance, tools such as the location-sharing systems Locaccino [153] and PeopleFinder [136] have been used to gain significant insight into the benefits of providing users with the ability to set personnal sharing policies. Two recurrent findings in UbiComp studies are that (i) users are not particularly good at effectively articulating their information-sharing policies (compared to their actual behavior) [136] and (ii) that sharing policies evolve over time [153, 136].

In order to overcome these two issues, machine learning techniques have been applied to improve to some extent the decision-making process [40, 53, 136]. The advantage of such systems is that they can decide in a (semi-)automatic fashion whether or not to share information. Most existing schemes, however, enable users to share only a specific kind of information (e.g., location). Moreover, they only make binary decisions on whether to share the requested information. In particular, this last issue is often mentioned as a crucial catalyst for overcoming concerns related to privacy [148] and for encouraging a more open, sharing behavior.

In this chapter, we perform a comprehensive study of information-sharing in mobile social networks by tackling, all at once, the issues related to context, user-burden trade-offs, and privacy. First, we develop a novel information-sharing system (SPISM) for (semi-)automatic decision-making in mobile social networks: It enables users to share different types of information (location, activity and co-presence of other people) with other users or services in a privacy-aware fashion. The decision-making core is supported by an active learning method that

[1]Social networking statistics, http://www.statisticbrain.com/social-networking-statistics/

enables SPISM to either decide automatically – whenever the confidence in the decision is high enough – or to rely on the user's input otherwise. Second, we conduct a personalized online study involving 70 participants where, in addition to collecting data about their sharing behaviors, we provide insight into two other crucial factors in UbiComp studies [10]: the *reason* behind a decision to share and the confidence that the user has in her decision. Third, we evaluate SPISM with respect to the amount of training data (provided by the user) and its performance, and we compare it against two policy-based mechanisms. Our results show that SPISM significantly outperforms both the individual user-privacy policies and several consolidated ones that are based on statistical analysis [3, 152], and it achieves up to 90% of correct sharing decisions. We also demonstrate the advantages of active learning techniques in our setting.

6.2 The SPISM Information-Sharing Platform

In this section, we describe the functionality, the operating principle, the architecture and the design of the SPISM information-sharing platform.

In order to better understand the following, we need to distinguish between two different kinds of subscribers to SPISM: (i) the *requester*, who wants to know something about other subscribers by sending information requests, and (ii) the *target*, who receives requests for information.

The SPISM platform is composed of the *SPISM application*, that runs on mobile devices (as for now it is implemented only for the Android platform), and the *SPISM Information Sharing Directory (ISD)*, that runs on a dedicated server.

6.2.1 Overview

The SPISM application enables subscribers, who can be users, third-party online services or mobile apps, to request information about other subscribers. The information that can be requested includes contextual data (the geographic location and the wireless identifiers of physically co-located devices) and the time-schedule availability. The features that are currently implemented are the following. The geographic location is determined by processing data obtained from the embedded GPS sensor (if available) or by WiFi tri-lateration (which relies on the Google localization service). The list of devices that are physically co-located with the target subscriber is obtained through periodic scans of the Bluetooth and WiFi interfaces. If a MAC address in the vicinity of the target is a known MAC address (there exist an entry associated with a subscriber in the contact list of the target), the name of the contact is displayed. Finally, the schedule availability is obtained from the subscriber's calendar (accessed through the on-device calendar application). Subscribers can specify a level of detail for the requested information: low, medium or high. The information sent by the target subscriber

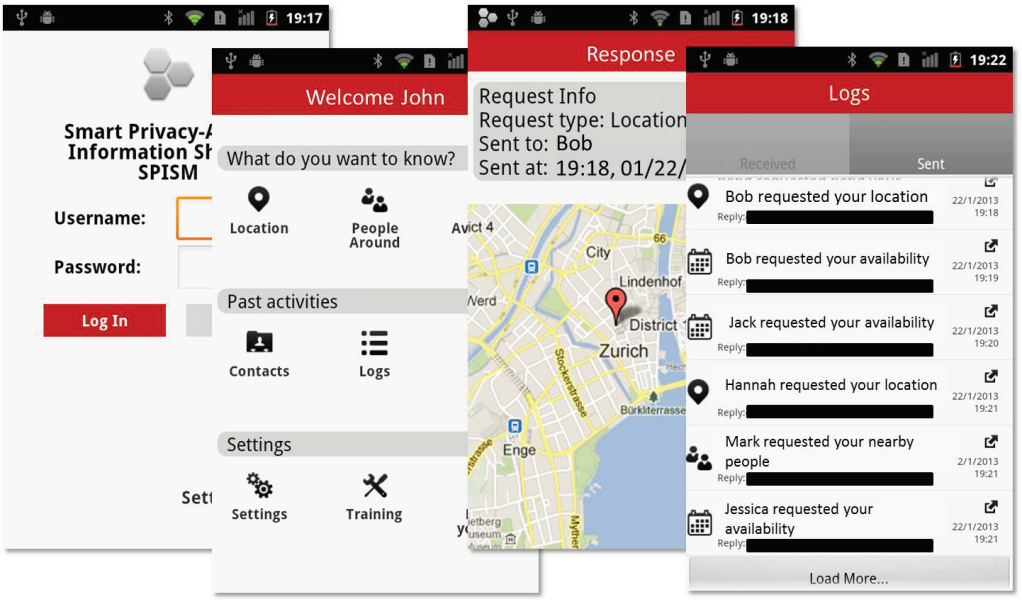


Figure 6.1: SPISM mobile application interfaces. From left to right, the different windows allow users to register and log in, check other subscribers' current location, the other devices around them, their availability. The subscribers can access other features such as the record of past activity and their contacts' lists.

is provided with a level of detail lower or equal to the requested level. For the location, the coordinates are truncated; for the neighboring devices, the presence (i.e., some devices/no devices), the number, or the identifiers of the devices are provided; for the schedule availability, the availability (i.e., busy/available), the title or the detailed record of the calendar activity is provided. Figure 6.1 shows the main application windows, where subscribers can log in and register, request the location, the co-located devices and the availability of their contact, as well as enjoy additional features such as visualizing the past activity and their contacts' list. At the time of this writing, we are implementing the possibility for users to audit and change a past decision, which will be then taken into account for the subsequent requests.

6.2.2 System Model

The SPISM platform is composed of the ISD and the subscribers of the service, who can be either users or third-party online services. The roles of the ISD and of the subscribers are as follows:

- **ISD**: Its main purpose is to allow users to discover the current IP addresses of their contacts when they want to send them information requests. The ISD stores the list of registered SPISM subscribers, their credentials, their

contact lists and the MAC addresses of the Bluetooth interfaces of each user's mobile devices. The subscribers interact with the ISD in the registration phase (once per user), during the log-in phase (once per application start), when downloading the contacts lists, when periodically reporting their IP addresses and updating their online status, and when sending information requests to one of their contacts.

- **Subscribers**: A subscriber, either an online service or a mobile user, can be a requester (when she sends queries to another subscriber) or a target (when she receives queries from other subscribers). In order to inform the ISD of her online status, each subscriber connected to the ISD sends periodic keep-alive messages. Requesters can see, at any time, the list of online and offline contacts, and they can choose to send queries to the online subscribers in their contacts list, in order to know their location, the devices around them and their availability. The requests that target subscribers receive and process are based on several features of their current physical and social contexts, including their current location, the time of the day and the people that are currently close by.

To enhance the security of the communications, all messages exchanged between the subscribers and the ISD are encrypted with a public-key certificate obtained from a trusted Certification Authority (CA). In order to protect users' privacy with respect to the ISD, no information requests or replies are tunneled through the ISD. This is a crucial aspect of our platform, as it prevents the service provider from learning the information sent by a subscriber about her location, physical context and availability. This approach has, however, two shortcomings as well: (i) The target knows the IP address of the requester (if no proxies are used), and therefore she may be able to infer the coarse-grained location of the requester (based on IP-geolocation), and (ii) the target may be able to infer the co-location of other users if they share the public IP address with the some of the requesters (when connected to a WiFi hotspot for example).

6.2.3 Operating Principle

SPISM works as follows. A user first logs in to the ISD with her username and password. She can subsequently report her online status and obtain the online status (and IP addresses) of her contacts from the ISD. In a typical scenario, the user requests some information from one of her (connected) contacts. To do so, the user first chooses the type of information she wants to request, by selecting the corresponding icon in the main window (See Figure 6.1), and then she selects the target subscriber from the list of her connected contacts. Finally, the user specifies the level of detail for the requested information and the request is prepared and sent directly to the target subscriber's device. If the reply is received within a

fixed amount of time (typically a few seconds) it is automatically showed to the user, together with the requested information if shared by the targeted requester (See Figure 6.1); otherwise, the user is redirected to the main window and she will be notified when the reply is received. At the targeted subscriber's device, the request is processed automatically when it is received: (1) The requested information is stored and (2) the information linked to the request (i.e., the time, the type of information requested and the requester) is combined with various contextual features (periodically collected in the background by SPISM from the various data sources and sensors available on the device) and fed to the information-sharing framework that we describe in detail in the next section. If SPISM can make the decision with enough confidence, based on the target subscriber's past decisions, the request is processed automatically. Otherwise, the target subscriber is notified and asked to decide; Her decision is then stored (note that the target subscriber can postpone her decision). Once a decision is made, it is sent back to the requester together with the requested information if the decision is positive. Before being sent, the requested information is processed to match the level of detail specified by the decision. All the sent and received requests are stored and can be accessed by the user by selecting the corresponding icon in the main window. In particular, the user can audit automatic decisions and correct those she disagrees with (to avoid similar errors in the future).

6.2.4 Decision Making

The SPISM information-sharing decision-making core processes each incoming information request. In order to make the decision, several contextual features are taken into account by the target device. Features such as the identity of and the social ties with the requester, the current location and the activity of the target, the people around the target and the time of the day were extensively studied in the past; several independent pieces of work show (with statistical significance) that they are strongly correlated with the information-sharing behavior of mobile users [152, 3, 27, 144, 35]. With these findings, we incorporated 18 such features in the SPISM decision-making core; the list of all the features we included is shown in Table 6.1. Due to the different natures of the features, some of them are defined as categorical (they are in a finite and pre-defined set of values, such as the social ties with the requester) or numerical (floating or integer values for the time and location coordinates).

Some of these 18 features can be extracted from the request itself or the target mobile device, such as the time, the current schedule availability or the requester ID, whereas other features require more information, e.g., the social ties with the requester and the semantics of the current location of the target subscriber. To obtain such information, SPISM takes advantage of the existing social networks, such as Facebook, and other data available on the phone (e.g., call logs). In ad-

	Feature	Type
Person	Familiarity	Float
	Social tie	Cat.
	User ID	Cat.
Service	Service category	Cat.
What?	Request type	Cat.
	Details	Float
Location	Latitude	Float
	Longitude	Float
	Semantic location	Cat.

	Feature	Type
When?	Time	Int.
	Weekday	Cat.
	Daytime	Cat.
	Activity	Cat.
With whom?	Neighbors	Int.
	Neighbors Type	Cat.
Last interact.	Time last request	Float
	Details last request	Float

Table 6.1: Features used by the SPISM machine learning framework to decide whether or not to share information and with what accuracy.

dition, other third-party services (such as Google Maps, OpenStreetMap and the Android application store, i.e., Google Play) are used to obtain more information about the location and type of application (in the case where the requester is a mobile application). In some cases, the extraction of the features requires access to the sensors embedded on the device; GPS and Bluetooth scans usually require a non-negligible amount of time and resources [130], and a per-request access to such sensors can drain the battery. For this reason, some time- or energy-consuming features (such as the GPS coordinates and Bluetooth MAC addresses of the nearby devices) are obtained periodically and cached, so that they can be polled by the device at any time instant without incurring resource-consuming operations. Note that the location, the list of nearby devices and the schedule availability are all used to make the decision and to be shared.

After all 18 features have been extracted from the request and determined from the context, they are aggregated into a feature vector and fed to a classifier. The output space of the classifier comprises a binary classifier that outputs a "share/not share" decision, followed by a linear regression function to estimate the level of detail. All machine learning components used by SPISM are implemented in the WEKA[2] Android library. Hereafter we detail the binary classifier and the linear regression components of SPISM.

[2]http://www.cs.waikato.ac.nz/ml/weka/

SPISM Stage I: To Share or not to Share

The first decision-making block of the SPISM framework consists of a logistic binary classifier, based on a logistic function defined as:

$$y_\omega(\phi(\mathbf{x})) = \frac{1}{1 + e^{-(\omega^{\mathbf{T}}\phi(\mathbf{x}))}} = \frac{1}{1 + e^{\sum_{i=1}^{M}\omega_i\phi_i(\mathbf{x})}} = p(S|\phi(\mathbf{x})) \tag{6.1}$$

where ω is the weight vector that is optimized during the training phase, M is the total number of elements of the feature map vector $\phi(\mathbf{x})$, S is the class that represents the "share" decision (whereas NS is the "not share" class), and $p(S|\phi(\mathbf{x}))$ is the probability of the "share" decision given the input feature map. We chose the logistic regression module for binary classification because (i) it does not penalize correct instances that are far from the classification hyperplane, (ii) it performs reasonably well with a small number (up to a few hundreds) of training instances and (iii) it outputs a value $\in [0, 1]$ that is also the posterior of the "sharing" class in our case.

At the first execution of the SPISM mobile application, this classifier is trained on a set of instances derived from the results of past research in information-sharing behavior [3, 152]. The sharing rules extracted from these works allow us to build a set of default sharing policies that reflect the tendencies that were observed with statistical significance. For example, people are usually less willing to share their information when they are sleeping or eating, and when they are at or close to their own home or to that of their friends. However, they are more likely to share when they are with friends and family, when they are alone or during the afternoon on weekdays (but not weekends).

After computing the probability of sharing $p(S|\phi(\mathbf{x}))$ for each received query, we distinguish between three possible outcomes:

$$p(S|\phi(\mathbf{x})) \begin{cases} > \tau_S & \text{then Share} \\ < \tau_{NS} & \text{then Not share} \\ \text{otherwise} & \text{Ask user} \end{cases}$$

where $0 \leq \tau_{NS} < \tau_S \leq 1$ are the thresholds for the confidence sharing decision that can be set for each subscriber. If $p(S|\phi(\mathbf{x})) > \tau_S$ then SPISM will automatically share the information, whereas if $p(S|\phi(\mathbf{x})) < \tau_{NS}$ the information will not be shared. In case SPISM is not confident enough to make an automated sharing decision (when $\tau_{NS} \leq p(S|\phi(\mathbf{x})) \leq \tau_S$), the subscriber will be prompted to manually decide whether she would like to share and to what extent. This weak form of active learning [140] is important as it allows the classifier's parameters to be recomputed each time there is a new manual sharing decision.

Currently, SPISM has three pre-defined levels of the confidence thresholds τ_S, τ_{NS} that can be selected by each subscriber: Low, medium and high confidence. The three levels of information-sharing thresholds have been adapted from

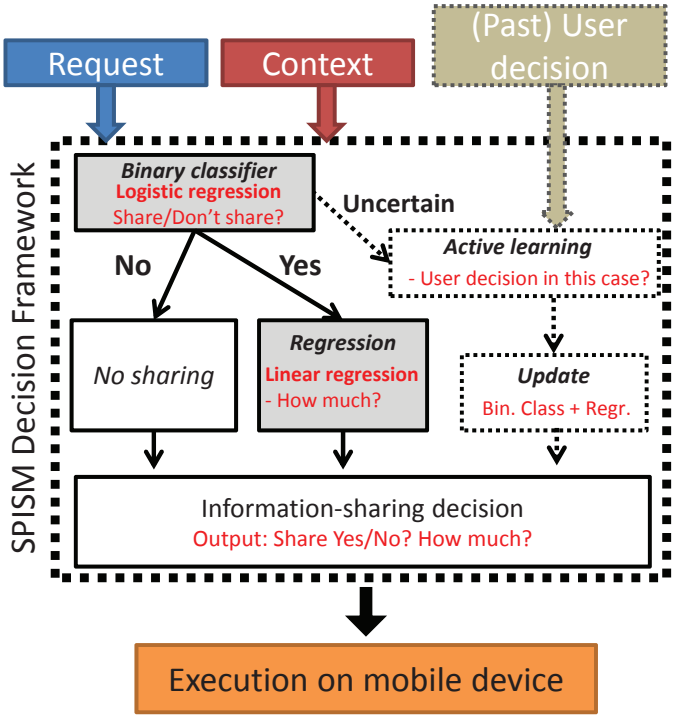


Figure 6.2: Diagram of the SPISM decision framework, which consists of a two-stage process that filters information requests and estimates the precision of the shared information to be included in the reply, based on the request, the context of the target and her past behavior.

prior works on the privacy attitudes of people in online scenarios [96, 121, 87]. When the low confidence threshold is selected ($\tau_{NS} = 0.45$, $\tau_S = 0.55$), SPISM will make most sharing decisions autonomously, without asking the subscriber, even when the confidence in the decision is quite low. The medium settings ($\tau_{NS} = 0.3$, $\tau_S = 0.7$) requires a slightly higher confidence in the decision in order not to ask the subscriber, whereas the high confidence settings ($\tau_{NS} = 0.1$, $\tau_S = 0.9$) would make the automated decision only if very confident, and would ask the subscriber otherwise. SPISM allows the subscriber to audit past sharing decisions (both automated and manual) and modify them for subsequent requests if she feels that the decisions do not reflect her preferences in that specific context. When a past decision instance is modified, the SPISM framework takes this modification into account for subsequent requests.

SPISM Stage II: How much to share?

If the decision of the first stage classifier is to "share", SPISM invokes the second stage which computes the extent of detail that is going to be shared, and consequently the level of obfuscation to be applied. For instance, subscribers

may want to share their location information at different resolutions with different requesters depending on the context [100]; they may want to share their geographic coordinates at the street, city or country level. Similarly, one may be willing to share the fact that she is currently not alone in some cases, whereas in other instances she may want to reveal the identity of the people nearby to the requester. Based on a linear regression function (Figure 6.2), SPISM determines the optimal accuracy of the information to be shared, according to the target's past behavior in similar contexts. After the obfuscation, the information is sent back to the requester.

6.3 Study and Data Collection

In order to better understand how users share information and to evaluate the efficacy of the SPISM framework with a large use sample, we ran an online user study in early 2013. The participants were not asked to use our proof-of-concept mobile application; however, their responses were fed to the actual decision-making core that is implemented in the application, so that the results of the study would reflect the actual behavior of the mobile application when making decisions. The study consists of an online survey that puts the participants in realistic, personalized and contextual UbiComp sharing scenarios where they are asked to answer a set of questions regarding their willingness to share private information, the confidence in and reason for their decisions.

6.3.1 Participants and Remuneration

We recruited people directly from four large university campuses (in the US, Canada and Europe), and indirectly via the Amazon Mechanical Turk platform (MTurk)[3]. The latter allowed us to draw participants from a pool of non-student population, in order to limit the bias towards academic and student behaviors. To advertise our study, we used dedicated mailing-lists and we ran a media campaign through Facebook, LinkedIn, Google+ and official university websites, coordinated by our academic media office. We screened participants according to the following prerequisites: (i) aged between 18 and 80 years, (ii) with an active Facebook account with at least 50 friends and (iii) uses a smartphone. Such criteria were selected so as to sample people that are active in social networks and are aware of the information-sharing possibilities linked to the use of smartphones. Furthermore, we screened the MTurk workers who could access our survey based on their past Human Intelligence Task (HIT) approval rate (>95%) and the number of past approved HITs (>100). This was only a preliminary step for preventing non-serious and inexperienced MTurk workers from accessing our survey.

[3]https://www.mturk.com/mturk/welcome

The survey requires access to private information of the participants (such as names of their friends on Facebook[4]) and it demands a significant amount of time (40 - 60 minutes). To provide incentives for the completion of the survey, we implemented two separate reward schemes: (i) the chance for one participant to win an Apple iPad and (ii) a fixed amount of money (US$4.5/HIT [106]). The first option was proposed to the participants recruited at the universities and through the academic media, whereas the second option was offered to the workers of the Amazon Mechanical Turk. We chose not to offer the second option to the academic participants due to our experience gained from previous on-campus studies: It appeared that the motivation for financial rewards was lower than for the possibility of winning a popular gadget.

6.3.2 Online Survey

We structured our survey in five parts: With a total of 94 questions, the first 19 are fixed (the same for each participant) and the last 75 are personalized (based on each participant's Facebook friends). In the very first part, the participants were required to log in to their Facebook account and grant our application access to their friend list.

In the first 15 questions, the participants were asked about their demographics, technology usage and privacy attitudes, in particular with respect to online social networks.

In the next question (16), the participants were asked to assign some of their friends to social groups, and we presented them with five distinct categories (based on [156]): (1) school colleagues, (2) friends, (3) family members, (4) work colleagues and (5) acquaintances. Each participant could assign one Facebook contact to at most one category. It is possible, however, that one such contact is a member of several categories (a school colleague that she works with currently). In this case, the participants were instructed to assign the contact to the most appropriate category.

In questions 17 through 19, the participants were asked to enter a set of information-sharing rules in free-text. The sharing rules are entered as a set of logical expressions that are based on the following *features*: (1) the participant's current location, (2) people nearby, (3) social group of the requester, (4) time of the day and (5) weekday/weekend. They can put *conditions* on these features (such as $=, <, >, \neq, \in$ or categorical values). For example, a location-sharing rule could be defined as:

[4]Before beginning the survey, the participants are informed that they would need to reveal the names of their Facebook friends for the purpose of this study. They approve a data retention and processing agreement, informing them that all data collected in our study is used solely for the purpose of our academic research project, and that we will not disclose or use it in any other way than what explicitly mentioned. Once the survey is completed, the name of the Facebook friends are replaced with anonymous identifiers.

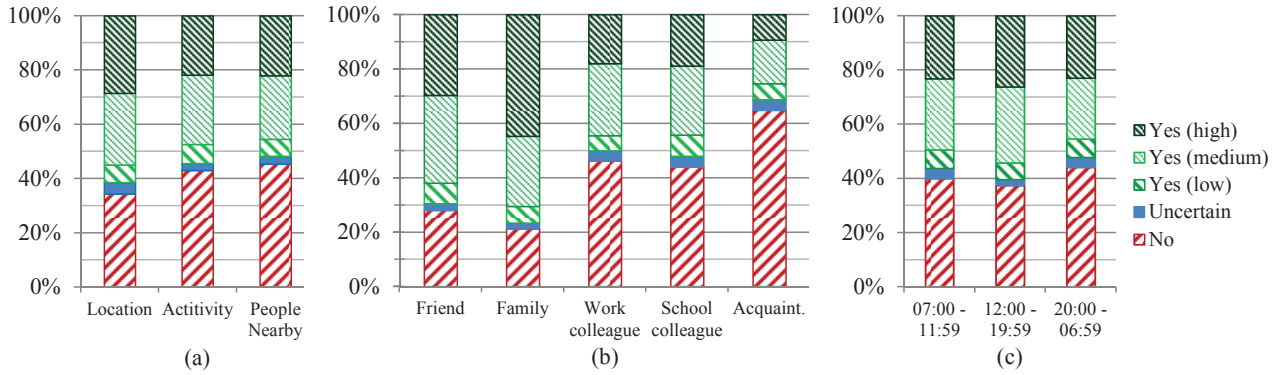


Figure 6.3: Histograms of the information-sharing decisions by (a) information type, (b) social group of the requester and (c) the time of the day.

> "I am at a *friend's place* **AND** with *acquaintance* **AND** the requester is a *work colleague*: do not share"

In the last 75 questions, the participants were presented with sharing scenarios and they were asked to decide whether they want to share the specific information in the given context, their confidence in the decision and the level of detail. A typical scenario is "Would you share your *location* with *John* on *Saturday* at *11:PM*, assuming you are at *an event* with *work colleagues*?" (where the requester name is chosen from the participant's Facebook friends and the other features are chosen at random). We use these six features (shown in italic in the previous sentence) for the classification task. In the actual mobile application, we will be able to use all 18 features as presented in Table 6.1.

Depending on their answers ("Yes", "No" and "Uncertain") to the questions in this part, participants were presented with sub-questions. More specifically, "Yes" and "No" answers were followed by a set of additional questions asking the participants about the confidence in their decisions (i.e., "not so confident", "confident", "very confident") and the features that influenced the most their decision (i.e., "requester", "day of the week", "time", "location" or "neighboring people"). For "Yes" answers, the participants were also asked about the level of detail of the shared information ("low", "medium" or "high"). Similarly, "Uncertain" answers were followed by sub-questions regarding the reasons for being uncertain, such as a conflict between some features (in this case, the participant can specify the features that motivates her the most to share and to not share, and then specify in free text the reason they conflict) or simply a lack of information (in this case the participant can specify which information would have helped her reach a decision).

In order to detect sloppy answers (e.g., random answers or bots), we included a number of "dummy" questions that require human understanding to be correctly

answered [106, 127]. These are questions such as simple computations (e.g., "3 + 4") or general-knowledge questions (e.g., "How many days are there in one week?"). Based on the answers to these questions and on the survey timing data (explained below), we ruled out dishonest participants from the dataset.

6.3.3 General Statistics and Validation

A total of 194 participants took part in our survey. 78 (40%) of them did not complete it, leaving 116 (60%) complete questionnaires. Out of these, 56 (48%) came from the university advertisement campaign (UNI) and 60 (52%) were recruited via MTurk. The average age of all the respondents is 27y±7 (Mturk avg. 31y±6, UNI avg. 25y±6), and 74% of them are male. 42% of all participants are students, 25% work in the IT industry and 8% in the education sector. It took 44±15 minutes on average to complete the survey (MTurk avg. 42 minutes, UNI avg. 47 minutes). We observed a sharp contrast, with respect to privacy concerns, between the two groups of participants: Most MTurk participants were not, or slightly, concerned about their privacy whereas most UNI participants were concerned about it.

Based on internal survey tests and detailed timing statistics, only the questionnaires that meet the following four validation criteria were retained.

- All answers to the dummy questions are correct;
- At least one different Facebook friend is assigned to each of the 5 social groups;
- The survey completion time is greater than 30 minutes.
- At least three of the following four timing conditions are met[5]: (1) Facebook friends assignment to groups time >5 minutes, (2) location sharing scenarios time >4 minutes, (3) activity sharing scenarios time >4 minutes, (4) nearby people sharing scenarios time >4 minutes.

All participants correctly answered the dummy questions. Based on timings, 46 (40%) of them were ruled out and 70 (60%) were kept for the analysis (33 MTurk and 37 UNI). The demographics remained mostly unaltered.

6.4 Analysis and Evaluation

In this section, we present three sets of results. First, using descriptive statistics of the survey questionnaire, we discuss the effect on the sharing decisions of different contextual features, of the requester, of the information type, and the

[5]These timing conditions were determined based on the observed timing distributions among all participants and on sample executions performed by test users.

main reasons behind the decisions. Second, we compare the performance of the SPISM automated decision-making process against that of the users' own policies and an established default policy. Third, we discuss the effects of the increase of user-involvement on the performance of SPISM, by using active learning with different confidence thresholds.

6.4.1 Survey Statistics

Based on the survey data, we computed the proportion of "Yes/No/Uncertain" decisions for the different values of each contextual feature we considered, such as the participant's current location, the social group of the requester, the time of day, day of week, and the type of information requested. We found that the two that have the largest effect on the decision are the social group of the requester and the type of information that is being requested.

Regarding the type of information being asked, Figure 6.3a shows that users disclose their location in 64% of the cases (the sum of the "yes (low)", "yes (medium)" and "yes (high)" bars, aggregated over the 70 participants and for all the 25 location-sharing questions – out of the 75 questions – that is a total of 1,750 answers), and only 8% of the time at a coarse granularity ("Yes (low)"). The information about activity and people nearby is disclosed 50% of the time. People tend to be slightly more willing to share their location than to share other information[6]: Location, contrary to the activity and the co-presence of other people, is widely shared information in most mobile social networks. In addition, this was confirmed by self-reported privacy concerns about information sharing on OSNs (not shown in the chapter).

Figure 6.3b shows the percentage of disclosure of information based on the social ties with the requester. We can see that, in accordance with previous Ubi-Comp studies, there are substantial differences[7] between the close ties ("family" and "friend") and the more distant ones ("acquaintances" and "colleagues"). For instance, the close ties are granted access to any type of information (70%-80%) more than twice the times compared to the more distant ones (30%). Moreover, the level of detail of the shared information is much higher for the close ties (up to 45% of "yes (high)") compared to the distant ones (down to 8%). In fact, the proportion of "Yes (low)" and "Yes (medium)" does not vary significantly. Hence, the results indicate that users tend to exhibit a more tailored sharing behavior depending on the type of information, the social ties and closeness with the requester[156]. As illustrated in Figure 6.3c, the time at which the request is sent does not substantially influence the decision: users are slightly less willing to share in the evening but exhibit the same behavior in the morning as in the

[6]With statistical significance, based on global and pair-wise χ^2 homogeneity tests with $p < 0.01$.

[7]Ibid.

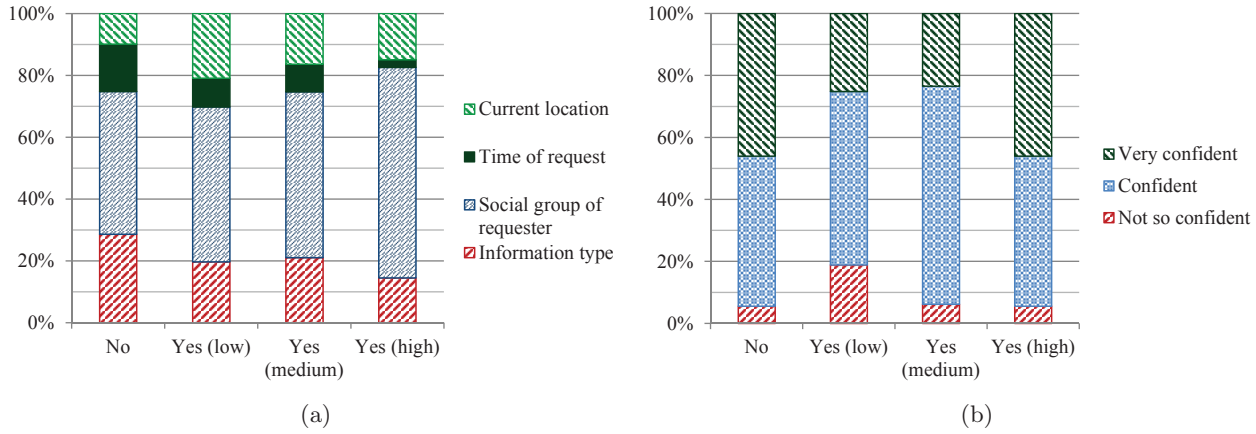


Figure 6.4: Histograms of users' responses to the survey questionnaire regarding (a) their main reasons for (not) sharing and (b) the confidence in their decisions.

afternoon[8]. Our findings are aligned with those obtained in [15], where the time of day and the location do not appear to be influential factors when sharing personal information such as location, as opposed to the type of social ties with the requester.

We also looked at the reasons for (not) disclosing information and at the users' confidence in their decisions. First we observe that the social ties with the requester is by far the most frequent reason for sharing (or not) information (45%-67%), followed by the type of information (15%-28%) and the current location (11%-21%). Second, we see again that the higher the level of detail (Figure 6.4(a)), the more important the social ties with the requester (on average). Unsurprisingly, the confidence that the participants have in their decision (Figure 6.4(b)) is lower for the intermediate level of detail: It can be observed that the proportion of "Very confident" is significantly lower for "low" and "medium" levels of detail than for "No" and "Yes (high)". In addition, the proportion of "Not so confident" is more than doubled for the most borderline decision, i.e., "Yes (low)". This could be explained by the fact that users try to minimize the risk by limiting the level of detail when their confidence is low.

6.4.2 Static Policies

We compared the performance of our SPISM decision framework with two other policy-based approaches. For the following comparisons, we used 10-fold cross validation and a logistic regression binary classifier. In order to be consistent with the policy-based approaches, we only compare the binary ("Yes/No") decisions here as the participants were instructed to only specify share/not share policies

[8]Ibid.

in the survey. The first policy-based approach, called *AT studies*, is inspired from the findings presented in [3, 152], and is derived by the following two rules:

1. Do not share any information while sleeping (12 AM - 6 AM) or eating (12 PM - 1 PM).

2. Do not share any information when you are around people that are not your family members or friends, except when you are at an event.

The second policy-based approach is derived from the individual policies that each participant specified in free text in the survey. We selected a random sample of 19 participants and we manually transposed their free-text policies to a format suitable to be evaluated against their own decisions. The participants specified between 1 and 15 policies (avg. 6.9).

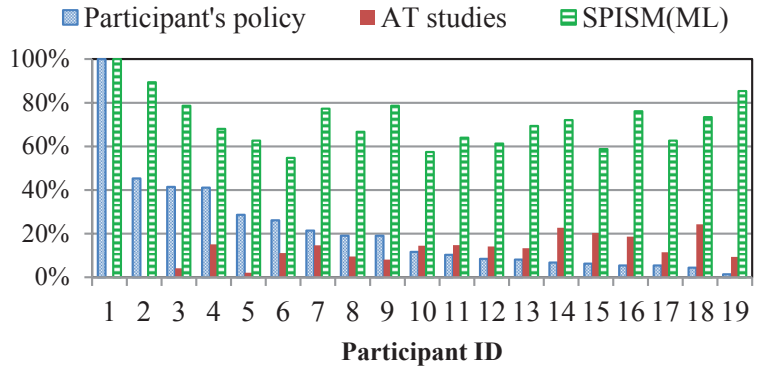
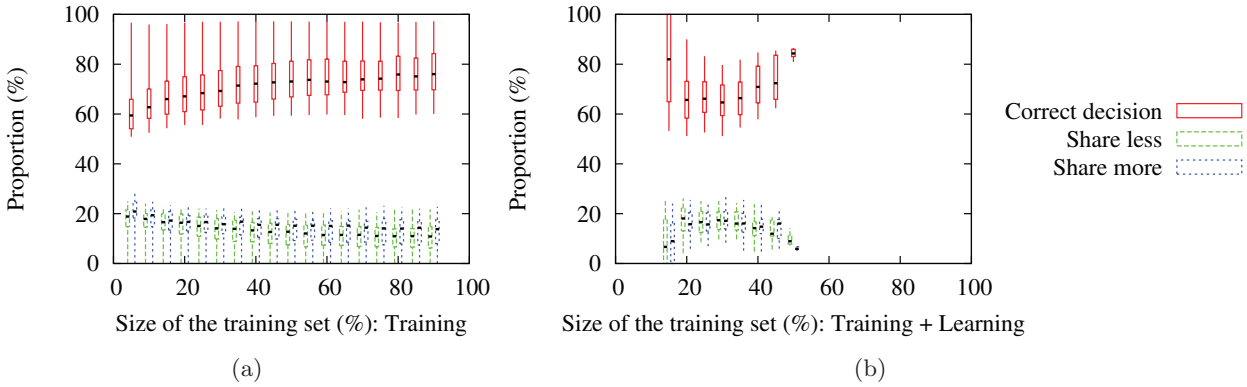


Figure 6.5: Histograms of the proportion of correct sharing decisions for three different sharing policy approaches. The AT studies' policies are derived from [3, 152], the participants' individual policies are derived from their free text answer in the survey and the SPISM approach is based on machine learning (without active learning).

The results of the three-way comparison are shown in Figure 6.5 where the results are sorted in descending order, based on the performance of the participant's individual policies. First, we can observe that the SPISM machine-learning approach consistently outperforms the other two approaches (this holds for all users when compared only to the AT policies defined earlier). The SPISM performance rate is between 53% and 100%, with an average of 71%. Compared to the participant's policy (avg. 22%) and the AT studies (avg. 12%), SPISM is significantly better at adapting itself to the user's sharing behavior. We also observe that usually where the participants' own policies correctly represent their actual behavior, the AT policies exhibit the worst performance (left side of Figure 6.5). The inverse appears to be true as well, as the policies inspired by the AT studies perform better for the participants whose own policies do not particularly match



(a) (b)

Figure 6.6: Performance of the machine learning-based decision making algorithm.

their actual behavior. This points out an interesting question, which is outside of the scope of this work: Are people who are not able to articulate well their sharing policy better suited to not trying to modify the default policies at all?

For the individual policies, we also observed the correctness of the decisions as a function of the number of policies, and found that a small number of policies (1-5) achieved up to 41% of correct decisions, followed by a slightly better performance for the number of policies between 6 and 9 (up to 45%), and then a much worse performance (up to 28% of correct decisions) for the highest number of policies (10 - 15). This suggests that there is an advantage in having a moderate number of sharing policies (up to 9) but not higher; With a larger number of policies, the risk of having overlapping but contradicting policies is higher, which could result in a worse overall performance.

6.4.3 Machine Learning

In order to assess the potential of (semi-)automatic information-sharing decision making, which constitutes the core of SPISM, we evaluate the performance of a logistic classifier in predicting the users' sharing decisions. To do so, we use the survey data comprised of 75 scenarios for each of the 70 participants: Each scenario corresponds to a feature vector and the decision made by the participant constitutes the ground truth. We considered only the "Yes" and "No" decisions. We evaluate the performance of the classifier in terms of the proportion of correct predictions (i.e., that match the user's decision), the proportion of cases where the information is incorrectly shared (whereas the user would have not shared it), thus compromising the user's privacy, and the proportion of cases where the information is incorrectly not shared (whereas the user would have shared it), thus reducing the utility of the system.

Firstly, we consider the case where the users first manually make n decisions to

train the classifier, and then the classifier makes the remaining decisions automatically. For several values of n, and for each participant, we compute the average proportions of correct and incorrect decisions following a 20-fold cross-validation approach. For each value of n, we obtain one data point (i.e., a proportion of "correct", "share less", and "share more" decisions) for each user and each fold, that is 1,400 data points. We represent the results across the different users and folds by showing the median, the first and third quartiles, and the 5 and 95-percentiles, as depicted in Figure 6.6(a). It can be observed that the median proportion of correct decisions increases from 60% and reaches 70% for a training set of only 30% of the data, which correspond to ∼25 scenarios. The proportion of correct decisions then quickly stabilizes around 74% after approximately 40 decisions (i.e., ∼50% of the data). The third quartile and the 95-percentile show that for more than 25% of the users, the proportion of correct decisions goes up to 80% and for some of them, it is consistently higher than 96%. The proportion of incorrect decisions is evenly distributed between sharing and not sharing the information, yet slightly biased towards incorrectly sharing the information. Should a user favor her privacy over the utility of the system, she could assign a higher error-penalty to this type of errors in order to make decisions in a conservative way. Without penalties and active learning, over-sharing happens in 10-20% of the cases, in line with the results reported in [15] and obtained with different classifiers. Note that the size of the training set (represented on the x-axis) represents the burden of the user as she has to manually make the corresponding decisions.

Secondly, we consider the case of active learning in which the user is asked to manually make the decision when the confidence of the classifier is low. The classifier outputs a distribution over the possible decisions; we define the confidence as the normalized entropy of this distribution. The classifier is first initialized with 10% of the data. For each user, we run the active learning-based classifier for several values of the confidence threshold (under which the user is asked to make the decision). Each experiment gives one data point comprised of (1) the proportion of decisions (including the first 10%) the user has to manually make and (2) the proportions of correct and incorrect decisions (among the decisions that are made automatically). In order to represent the data in a form that is comparable to that of Figure 6.6(a), we group the data points in bins of size 5% (on the x-axis as represented in the figure) based on the proportion of manual decisions. Note that the number of data points varies across the different bins. Within each bin, we compute the median and the relevant percentiles. The result are depicted in Figure 6.6(b). It can be observed that active learning outperforms training-only learning in most cases (i.e., for a given number of manual decisions, it provides a higher proportion of correct decisions). The proportion of manual decisions remains lower than 50% which shows that the classifier can

make the decision with very high-confidence for at least half of the scenarios. For some users, the proportion of manual decisions remains low (∼20%), regardless of the confidence threshold, and the proportion of correct decisions is high (∼80%). This corresponds to the users whose decisions are highly predictable. With active learning, we observe a significantly improved performance in terms of over-sharing compared to the absence of active learning. We posit that, coupled with cost-sensitive classifiers, active learning can be used to improve the correctness of the sharing decisions while maintaining a significantly lower over-sharing rate.

6.5 Related Work

A substantial research effort has been made on the topic of privacy and information sharing in mobile social networks, notably with respect to the attitudes of people when sharing static and contextual data with other peers. The studies that are most related to our work can be grouped, from a high-level perspective, into two categories: (i) contextual information sharing and privacy [144, 152, 12] and (ii) machine learning for information sharing [40, 53, 133, 112, 2, 136].

Contextual Information Sharing and Privacy Smith et al. [144] provide an early investigation on technologies that allow people to share their contextual information, such as location, in mobile social networks. In addition to allowing users to manually decide when to share their location with others, the authors implemented a system called *Reno* that can automate the process based on a set of pre-defined regions. By allowing Reno to automatically send notifications whenever the user entered or exited such regions, the authors show that there is both a value and a cost associated with automatic information disclosure. In particular, they show that static rules for location sharing in pre-defined regions are ineffective in accurately expressing the users' actual behavior when other contextual elements change, such as the time of the day or the day of the week.

More recently, Toch et al. [152] study the effect of the type of locations visited by the users on their willingness to share them with others. By considering simple statistical models that take into account factors other than the geographic location, the authors showed that the semantic category of the location being shared (such as a shopping center or a hospital) and the social group of the person asking for the location are significant factors in deciding whether to share the location. These results support earlier efforts [3, 104, 136] in providing a set of contextual features that have a statistically significant impact on the location-sharing behavior of mobile users.

In an attempt to capture the cost of mistakenly revealing a location due to ineffective sharing policies, in addition to sharing preferences, Benisch et al.

[12] compare simple access control policies (white lists) to more sophisticated ones (based on time, day and location). They found out that (i) the accuracy of the sharing policies increases with their complexity (or flexibility), and that (ii) the accuracy benefits are the greatest for the highly sensitive information. This suggests that the notion of the cost of mistakenly revealing information to unauthorized parties (in particular contexts) is an important factor in designing and optimizing automated information-sharing mechanisms.

Wiese et al. [156] investigate the effect of physical and perceived social closeness on people's willingness to share information with others. Among the main results of the study, the authors show that social closeness and the frequency of communication are better predictors of sharing than physical proximity. Moreover, these two factors were also shown to have a capacity to predict sharing better than the social groups of the people asking for the information. Thus, the authors suggest that automatic methods for inferring social closeness could be suited for accurate information-sharing decisions more than physical co-location, in the case automated mechanisms (such as in [147, 84, 149, 113]) are envisaged.

Machine Learning and Information Sharing Whereas studies on information-sharing attitudes and privacy shed light on the behavior of people and the factors that influence their decisions, they are mostly concerned about understanding the causes and effects of such behavior. Meanwhile, there has been a substantial effort in devising methods that help and nudge the users to make information-sharing decisions, or even make decisions on their behalf. We present some of these methods, including both supervised and unsupervised approaches for decision-making.

In [136], Sadeh et al. compare the accuracy of user-defined sharing policies with an automated mechanism (case-based reasoner) and a machine learning approach (random forests), showing that these approaches have an accuracy better than the user-defined policies. Owing in part to the greater flexibility of the supervised machine-learning approaches compared to the more coarse-grained user-defined policies, the automated methods also benefited from the fact that users appeared to not be able to create sharing rules consistent with their own choices. On the contrary, the feedback provided by the users to the machine-learning methods did however appear to be consistent with their actual sharing behavior, which helped the automated methods to achieve better accuracy results.

Unsupervised or semi-supervised methods, which reduce the initial setup burden of the default sharing policies for each user, are investigated in [40, 53]. For instance, Danezis [40] proposes a method for automatically extracting privacy settings for online social networks; the method is based on the notion of a limited proliferation of information outside of a given social context. The proposed method, which determines cohesive groups of users where users belonging to a

group have stronger ties to the users outside of the group, shows promising results on a limited set of evaluation samples. This study also shows that the social groups, and especially methods for their automated extraction, are a key factor to sharing private information in social networks.

Fang and LeFevre [53] propose a novel approach to the inference and definition of access control policies for personal information on online social networks. They enable the supervised learning mechanism to learn the sharing preferences of a user by asking her a limited number of questions about her sharing behavior with some of her friends; these specific friends are the most "informative", i.e., those for which the classifier is most uncertain about. The authors show that their approach of iteratively asking questions about the most uncertain case (active learning with uncertainty sampling) reduces the effort required by the users and maintains a high accuracy compared to the ground truth (based on a 45-user study on Facebook).

Bigwood et al. [15] evaluate different machine learning algorithms for information sharing in terms of information over-exposure and correct decisions. Although their work is focused exclusively on binary (yes/no) location-sharing, the authors provide a machine-learning-based determination of the most influential features for the sharing decisions; moreover, they take into account cost-sensitive classifiers to reduce over-exposure.

An interesting approach towards the assessment of the privacy risks in online social networks is proposed by Liu and Terzi [102], who introduced a novel method for computing the privacy score of a user. Based on two values, the *sensitivity* and the *visibility* of the personal information, their score captures both the role of the user's own preferences for sharing (sensitivity of the released information to the individual) and her role (or visibility) in the network; the greater the sensitivity of the information item, the higher is her privacy score. Similarly, the greater the number of people who know a particular information item about a user (visibility of the information), the higher is her privacy score. Furthermore, when coupled with probability estimation techniques used in Item-Response Theory (IRT) [103], such privacy scores can be comparable across different social networks (such as Facebook, MySpace, Twitter). The latter property is important whenever users may have online accounts on different social networks, as it may be possible to link information shared on different social networks to reconstruct the information about the user [114].

6.6 Summary

Mobile social networks enable users to share an increasing number of contextual information, such as their location, their activity and their co-presence with others. To simplify the sharing process and improve usability, the research com-

munity has been studying sharing preferences and developing applications that, based on several contextual features, can automate to some extent the sharing process. Machine-learning approaches have been developed and evaluated for specific instances of information (mostly location) or for online social network (without the notion of context).

In this chapter, we have presented and evaluated a novel privacy-preserving information-sharing system (SPISM) that decides in a (semi-)automated fashion whether or not to share different types of contextual information and to what level of detail. Using a personalized online user-study involving 70 participants, we show that SPISM significantly outperforms both individual and general user-defined sharing policies, achieving up to 90% of correct sharing decisions, with only a limited cost for the user in terms of initial setup thanks to active learning. We also show that the system has a slight bias towards incorrectly sharing the information, which could be mitigated by introducing a penalty for this kind of errors. Furthermore, our results provide significant insight into two other crucial aspects of UbiComp studies: the reasons behind sharing decisions and the participants' confidence in them. We show that the type of the requested information, in addition to the social ties of the requester, is an influential feature in the decision process.

Publication: [16]

Chapter 7

Conclusion

In this thesis, we have focused on the privacy of mobile users in context-aware networks. From the application to the link layer of the ISO/OSI stack, we have performed information leakage analysis and designed privacy-preserving protocols in order to enable users to benefit from added security and privacy while executing common tasks and using social and context-aware services. In our studies, we involved a large population of external participants in both online and in-situ experiments, in order to assess the challenges they face with respect to privacy and to evaluate our protocols.

In Part I, we have studied two privacy problems that arise on the application-layer: meeting scheduling and optimal meeting location determination for mobile devices. For these two problems, we designed and evaluated privacy-preserving protocols that enable users to privately determine common meeting times and to choose, among a set of private preferred places, the optimal one among them. By means of targeted user studies and prototype applications, we have demonstrated that our protocols can be efficiently run on current mobile devices and that they fare better, with respect to computational and communication complexity, compared to existing centralized and distributed approaches. Furthermore, our proof-of-concept applications scored well with the users, who found them easy to use and provided us with input to formulate design guidelines for application developers.

In Part II, we have investigated the issues of social community and relationship privacy in pervasive networks. First, we conducted the first large experiment in which we compared the accuracy in inferring social communities of an external eavesdropper, who did not have access to the data stored on the mobile devices, with that of a malicious application or an experimenter who had access to such data. Our results showed that the eavesdropper could infer, with high accuracy, generic statistics of social communities of users, even by using a basic

characterization for the weight functions of the underlying social graph of users' encounters. With a more comprehensive weight function, such an adversary can achieve positive results also with respect to the identity of the members of the social communities. Moreover, we have studied the effect of exploiting the inferred social community information in reconstructing the social ties of the mobile users. In particular, we have shown how the false positives can be reduced by a factor of three while doubling the true positive rate of social ties inference for specific types of relationships. By adjusting the inference parameters, it is possible to achieve high success rates for specific target social relationship classes.

In Part III, we have focused on cross-layer approaches for privacy protection in mobile social networks. The context-awareness of current mobile devices and the possibility of sharing such information with almost anyone on the social network represent two important aspects to consider when developing privacy-aware information-sharing mechanisms. We designed a novel and automated information-sharing mechanism that enables users to share different kinds of personal and contextual information with each other and third parties. Our decision-making core is developed with both privacy and ease of use in mind, as it mimics the users' behavior with respect to information sharing – by learning from the past behavior and the current context – and it relieves the users from having to decide manually and once and for all whether they want to share personal information. In addition to the sharing decision, our system is also able to infer the level of granularity for each information request. Our results indicate that our proposed mechanism is able to make better sharing decisions than both the existing systems and users' own sharing policies. Furthermore, by choosing the appropriate questions when user input is required – by actively choosing the question that yields the most information to the classifier – we can minimize the number of interruptions for the users while maximizing the gain in terms of confidence in the decision by the mobile device.

Bibliography

[1] H. Akaike. A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, 19:716–723, 1974. [cited at p. 116]

[2] X. An, D. Jutla, N. Cercone, C. Pluempitiwiriyawej, and H. Wang. Uncertain Inference Control in Privacy Protection. *International Journal of Information Security*, 8(6):423431, 2009. [cited at p. 145]

[3] D. Anthony, T. Henderson, and D. Kotz. Privacy in Location-Aware Computing Environments. *IEEE Pervasive Computing*, 6(4):64, 2007. [cited at p. 129, 132, 134, 142, 145]

[4] Apple AirDrop. `http://www.apple.com/macosx/whats-new/`, Visited on 01/Jul/2012. [cited at p. 88]

[5] Apple iCal. http://apple.com/ical, Last visited 27.01.2011. [cited at p. 12]

[6] C.A. Ardagna, M. Cremonini, E. Damiani, S. Vimercati, and P. Samarati. Location privacy protection through obfuscation-based techniques. In *21st Annual IFIP WG 11.3 working conference on data and applications security*, 2007. [cited at p. 83]

[7] C. Arthur. iphone keeps record of everywhere you go. The Guardian, UK, April 2011. [cited at p. 3]

[8] Aziala-net. http://icawww1.epfl.ch/aziala/index.html. [cited at p. 90]

[9] J. Banford and J. Irvine. Estimating social graphs in an education environment. *IEEE Vehicular Technology Magazine*, 7(1), 2012. [cited at p. 123]

[10] L. Barkhuus. The mismeasurement of privacy: using contextual integrity to reconsider privacy in HCI. In *ACM Conference on Human Factors in Computing Systems*, pages 367–376, 2012. [cited at p. 129]

[11] M. Bellare. A note on negligible functions. *Journal of Cryptology*, 15:271–284, 2008. 10.1007/s00145-002-0116-x. [cited at p. 25]

[12] M. Benisch, P. G. Kelley, N. Sadeh, and L. F. Cranor. Capturing Location-Privacy Preferences: Quantifying Accuracy and User-Burden Tradeoffs. *Personal and Ubiquitous Computing*, 15(7):679–694, 2011. [cited at p. 128, 145, 146]

[13] A.R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2, 2003. [cited at p. 103, 121, 123]

[14] F. Berger, R. Klein, D. Nussbaum, J.R. Sack, and J. Yi. A meeting scheduling problem respecting time and space. *Geoinformatica*, 13(4):453–481, 2009. [cited at p. 82]

[15] G. Bigwood, F. B. Abdesslem, and T. Henderson. Predicting Location-Sharing Privacy Preferences in Social Network Applications. In *Workshop on recent advances in behavior prediction and pro-active pervasive computing*, 2012. [cited at p. 141, 144, 147]

[16] I. Bilogrevic, K. Huguenin, B. Agir, J.-P. Hubaux, and M. Jadliwala. Adaptive information-sharing for privacy-aware mobile social networks. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2013. [cited at p. 2, 7, 148]

[17] I. Bilogrevic, K. Huguenin, M. Jadliwala, F. Lopez, J.-P. Hubaux, P. Ginzboorg, and V. Niemi. Inferring social ties in academic networks using short-range wireless communications. In *Workshop on Privacy in the Electronic Society*, pages 1–10, 2013. [cited at p. 7, 124]

[18] I. Bilogrevic, M. Jadliwala, J.-P. Hubaux, I. Aad, and V. Niemi. Privacy-preserving activity scheduling on mobile devices. In *ACM Conference on Data and Applications Security and Privacy*, pages 261–272. ACM, 2011. [cited at p. 7, 49]

[19] I. Bilogrevic, M. Jadliwala, V. Joneja, K. Kalkan, J.-P. Hubaux, and I. Aad. Privacy-preserving location sharing to determine optimal meeting locations on mobile devices. *IEEE Transactions on Information Forensics and Security*, (under submission). [cited at p. 7, 84]

[20] I. Bilogrevic, M. Jadliwala, K. Kalkan, J.-P. Hubaux, and I. Aad. Privacy in mobile computing for location-sharing-based services. In *Privacy Enhancing Technologies Symposium*, pages 77–96. Springer, 2011. [cited at p. 7, 53, 84]

[21] I. Bilogrevic, M. Jadliwala, P. Kumar, S. S. Walia, J.-P. Hubaux, I. Aad, and V. Niemi. Meetings through the cloud: Privacy-preserving scheduling on mobile devices. *Journal of Systems and Software*, 84(11):1910–1927, 2011. [cited at p. 7, 49]

[22] I. Bilogrevic, M. Jadliwala, I. Lám, I. Aad, P. Ginzboorg, V. Niemi, L. Bindschaedler, and J.-P. Hubaux. Big brother knows your friends: on privacy of social communities in pervasive networks. In *International Conference on Pervasive Computing*, pages 370–387. Springer, 2012. [cited at p. 2, 7, 104, 109, 123]

[23] L. Bindschaedler, M. Jadliwala, I. Bilogrevic, I. Aad, P. Ginzboorg, V. Niemi, and J.-P. Hubaux. Track me if you can: on the effectiveness of context-based identifier changes in deployed mobile networks. In *Network and Distributed System Security Symposium*, 2012. [cited at p. 93, 123]

[24] D. Boneh, E. Goh, and K. Nissim. Evaluating 2-dnf formulas on ciphertexts. In *Theory of cryptography*, pages 325–341. Springer, 2005. [cited at p. 59, 60, 74]

[25] Stephen P. Borgatti, Ajay Mehra, Daniel J. Brass, and Giuseppe Labianca. Network Analysis in the Social Sciences. *Science*, 323:892–895, 2009. [cited at p. 110]

[26] A. Bose and C.H. Foh. A Practical Path Loss Model for Indoor WiFi Positioning Enhancement. In *International Conference on Information, Communication and Signal Processing*, 2007. [cited at p. 96]

[27] A. J. Brush, J. Krumm, and J. Scott. Exploring End User Preferences for Location Obfuscation, Location-Based Services, and the Value of Location. In *International Conference on Ubiquitous Computing*, 2010. [cited at p. 132]

[28] Business Week. Facebooks value tops amazon.com; trails only google on web. `http://www.businessweek.com/news/2011-01-28/facebook-s-value-tops-amazon-com-trails-only-google-on-web.html`, Visited on 01/Jul/2012. [cited at p. 88]

[29] C. Cachin and R. Strobl. Asynchronous group key exchange with failures. In *ACM Symposium on Principles of Distributed Computing*, pages 357–366, New York, NY, USA, 2004. ACM. [cited at p. 14, 54]

[30] B. Carbunar, R. Sion, R. Potharaju, and M. Ehsan. The shy mayor: Private badges in geosocial networks. In *International Conference on Applied Cryptography and Network Security*, 2012. [cited at p. 83]

[31] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of Human Mobility on Opportunistic Forwarding Algorithms. *IEEE Transactions on Mobile Computing*, 6:606–620, 2007. [cited at p. 88, 103]

[32] C.-H. Owen Chen, C.-W. Chen, C. Kuo, Y.-H. Lai, J. M. McCune, A. Studer, A. Perrig, B.-Y. Yang, and T.-C. Wu. Gangs: Gather, authenticate 'n group securely. In *International Conference on Mobile Computing and Networking*, pages 92–103, New York, NY, USA, 2008. ACM. [cited at p. 14, 54]

[33] M.H. Chignell, A. Quan-Haase, and J. Gwizdka. The privacy attitudes questionnaire (paq): initial development and validation. In *Human Factors and Ergonomics Society Annual Meeting*, volume 47, pages 1326–1330. 2003. [cited at p. 39, 78]

[34] CHIlabs PDA (Personal Digital Assistants) Use Study. http://personal.bgsu.edu/˜nberg/chilabs/pda.htm, Visited Mar. 2011. [cited at p. 12]

[35] S. Consolvo, I.E. Smith, T. Matthews, A. LaMarca, J. Tabert, and P. Powledge. Location Disclosure to Social Relations: Why, When, & What People Want to Share. In *ACM SIGCHI Conference on Human Factors in Computing Systems*, 2005. [cited at p. 132]

[36] Nokia Conversations. Nokia instant community gets you social, 2010. [cited at p. 88, 90, 106]

[37] M.S. Corson, R. Laroia, J. Li, V. Park, T. Richardson, and G. Tsirtsis. Toward Proximity-aware Internetworking. *Wireless Communications*, 2010. [cited at p. 88]

[38] D.J. Crandall, L. Backstrom, D. Cosley, S. Suri, D. Huttenlocher, and J. Kleinberg. Inferring social ties from geographic coincidences. *National Academy of Sciences*, 107, 2010. [cited at p. 2, 88, 103, 106, 123]

[39] Dailywireless.org. http://www.dailywireless.org/2009/03/24/smartphone-users-100m-by-2013, Last visited 27.01.2011. [cited at p. 12]

[40] G. Danezis. Inferring Privacy Policies for Social Networking Services. In *ACM Workshop on Artificial Intelligence and Security*, 2009. [cited at p. 128, 145, 146]

[41] E. De Cristofaro and G Tsudik. Practical private set intersection protocols with linear complexity. *Financial Cryptography and Data Security*, 2010. [cited at p. 12, 47]

[42] I. Derényi, G. Palla, and T. Vicsek. Clique percolation in random networks. *Physical review letters*, 94, 2005. [cited at p. 97]

[43] C. Diot. Haggle Project. http://www.haggleproject.org/. Last visited 13/Mar/2013. [cited at p. 123]

[44] D. Dolev and A. Yao. On the Security of Public Key Protocols. *IEEE Transactions on Information Theory*, 29, 1983. [cited at p. 91]

[45] Doodle: easy scheduling. http://www.doodle.com/, Visited Nov. 2011. [cited at p. 12, 46]

[46] W. Du and M.J. Atallah. Secure multi-party computation problems and their applications: a review and open problems. In *Workshop on New security paradigms*, pages 13–22. ACM New York, NY, USA, 2001. [cited at p. 12, 47, 48]

[47] N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10:255–268, 2006. [cited at p. 89, 106, 112, 123]

[48] N. Eagle, A.S. Pentland, and D. Lazer. Inferring friendship network structure by using mobile phone data. *National Academy of Sciences*, 106, 2009. [cited at p. 88, 103, 106, 123]

[49] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985. [cited at p. 16, 22, 59, 60]

[50] E. Ephrati, G. Zlotkin, and J. S. Rosenschein. Meet your destiny: A non-manipulable meeting scheduler. In *ACM conference on Computer Supported Cooperative Work*, pages 359–371, New York, NY, USA, 1994. ACM. [cited at p. 13]

[51] Facebook Deals. http://www.facebook.com/deals/, Visited Nov. 2011. [cited at p. 52]

[52] Facebook Statistics. http://www.facebook.com/press/info.php?statistics, Visited Nov. 2011. [cited at p. 52]

[53] L. Fang and K. Lefevre. Privacy Wizards for Social Networking Sites. In *World Wide Web Conference*, 2010. [cited at p. 128, 145, 146, 147]

[54] Femto Forum. http://femtoforum.org/fem2/pressreleases.php ?id=277, Last visited 13/Mar/2013. [cited at p. 88, 89, 121]

[55] M. Follman. "Bluetoothing" Iran's revolution. *Markfollman.com*, 2010. [cited at p. 88]

[56] FON. http://corp.fon.com/en. Last visited 13/Mar/2013. [cited at p. 121]

[57] S. Fortunato. Community detection in graphs. *Physics Reports*, 486, 2010. [cited at p. 92]

[58] Foursquare for Business. http://foursquare.com/business/, Visited Nov. 2011. [cited at p. 52]

[59] M.S. Franzin, EC Freuder, F. Rossi, and R. Wallace. Multi-agent meeting scheduling with preferences: Efficiency, privacy loss, and solution quality. *Computational Intelligence*, 20(2), 2004. [cited at p. 13]

[60] Map of FreeWifi Hotspots. `http://iphone.r0ro.fr/fwConnect/map.php`, Visited on 01/Nov/2012. [cited at p. 122]

[61] J. Freudiger, M. Jadliwala, J.-P. Hubaux, V. Niemi, and P. Ginzboorg. Privacy of community pseudonyms in wireless peer-to-peer networks. *Mobile Networks and Applications*, pages 1–16, 2011. [cited at p. 52]

[62] J. Freudiger, R. Shokri, and J.-P. Hubaux. Evaluating the privacy risk of location-based services. In *Financial Cryptography and Data Security*, pages 31–46. Springer, 2012. [cited at p. 52]

[63] Keith B. Frikken and Mikhail J. Atallah. Privacy preserving route planning. In *Workshop on Privacy in the Electronic Society*, pages 8–15. 2004. [cited at p. 82]

[64] B. Gedik and L. Liu. Location Privacy in Mobile Systems: A personalized Anonymization Model. In *IEEE International Conference on Distributed Computing Systems*, 2005. [cited at p. 83]

[65] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.L. Tan. Private queries in location based services: anonymizers are not necessary. In *ACM Special Interest Group on Management Of Data*, pages 121–132. 2008. [cited at p. 55]

[66] Goldman Sachs. Clash of the titans. http://www.technologyinvestor.com/wp-content/uploads/2012/12/AAPL-by-Goldman-Sachs.pdf, 12 2012. [cited at p. 1]

[67] O. Goldreich. *Foundations of Cryptography*, volume 1. Cambridge University Press, 2001. [cited at p. 14, 19]

[68] O. Goldreich. *Foundations of cryptography: Basic applications*. Cambridge University Press, 2004. [cited at p. 55, 68]

[69] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984. [cited at p. 16, 29, 47]

[70] P. Golle and K. Partridge. On the anonymity of home/work location pairs. In *Pervasive Computing*. 2009. [cited at p. 52]

[71] N.W. Gong, M. Laibowitz, and J. Paradiso. Dynamic Privacy Management in Pervasive Sensor Networks. In *International Joint Conference on Ambient Intelligence*, 2010. [cited at p. 89]

[72] MC González, HJ Herrmann, J. Kertész, and T. Vicsek. Community structure and ethnic preferences in school friendship networks. *Physica A: Statistical Mechanics and its Applications*, 379, 2007. [cited at p. 88, 103]

[73] Google Now. http://www.google.com/landing/now/, Last visited 14/05.2013. [cited at p. 1]

[74] Google Smart Rescheduler. http://gmailblog.blogspot.com/2010/03/smart-rescheduler-in-google-calendar.html, Last visited 27.01.2011. [cited at p. 12]

[75] M. Gruteser and B. Hoh. On the Anonymity of Periodic Location Samples. In *International Conference on Security in Pervasive Computing*, 2005. [cited at p. 2, 88]

[76] Saikat Guha, Mudit Jain, and Venkata Padmanabhan. Koi: a location-privacy platform for smartphone apps. In *Symposium on Networked Systems Design and Implementation*, 2012. [cited at p. 83]

[77] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning (2nd Edition)*. Springer, 2008. [cited at p. 92]

[78] T. Henderson, D. Kotz, and I. Abyzov. The changing usage of a mature campus-wide wireless network. In *International Conference on Mobile Computing and Networking*, 2004. [cited at p. 89]

[79] T. Herlea, J. Claessens, B. Preneel, G. Neven, F. Piessens, and B. De Decker. On securely scheduling a meeting. In *IFIP TC11 16th International Conference on Information Security (IFIP/Sec'01)*, pages 183–198. Kluwer Academic Pub, 2001. [cited at p. 12, 29, 35, 36, 47, 48]

[80] M. Herrmann, A. Rial, C. Diaz, and B. Preneel. Privacy-preserving location-sharing-based services. Technical report, COSIC, KU Leuven, 2013. [cited at p. 83]

[81] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Enhancing Security and Privacy in Traffic-Monitoring Systems. *IEEE Pervasive Computing*, 5:38–46, 2006. [cited at p. 2, 88, 106]

[82] J.I. Hong and J.A. Landay. An architecture for privacy-sensitive ubiquitous computing. In *Conference on Mobile Systems, Applications, and Services*, 2004. [cited at p. 103]

[83] T. Hossmann, T. Spyropoulos, and F. Legendre. Know thy neighbor: Towards optimal mapping of contacts to social graphs for dtn routing. In *International Conference on Computer Communications*, 2010. [cited at p. 88, 103]

[84] G. Hsieh, K. P. Tang, W. Y. Low, and J. I. Hong. Field Deployment of IMBuddy: A Study of Privacy Control and Feedback Mechanisms for Contextual IM. In *ACM International Conference on Ubiquitous Computing*, 2007. [cited at p. 146]

[85] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket switched networks and human mobility in conference environments. In *ACM SIGCOMM workshop on Delat-Tolerant Networks*, 2005. [cited at p. 88, 103]

[86] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. *IEEE Transactions on Mobile Computing*, 2010. [cited at p. 88, 103]

[87] G. Iachello and J. Hong. End-User Privacy in Human-Computer Interaction. *Foundations and Trends in Human-Computer Interaction*, 1(1):1–137, 2007. [cited at p. 135]

[88] ITU Statistics. http://www.itu.int/ict/statistics. [cited at p. 88]

[89] P. Jaccard. Etude comparative de la distribution florale dans une portion des alpes et du jura, 1901. [cited at p. 99]

[90] M. Jadliwala, I. Bilogrevic, and J.P. Hubaux. Optimizing mixing in pervasive networks: A graph-theoretic perspective. *European Symposium on Research in Computer Security*, 2011. [cited at p. 103]

[91] S. Jaiswal and A. Nandi. Trust no one: A decentralized matching service for privacy in location based services. In *ACM SOSP Workshop on Networking, Systems, and Applications on Mobile Handhelds*, 2010. [cited at p. 82]

[92] B. Kellermann and R. Böhme. Privacy-enhanced event scheduling. In *IEEE International Conference on Computational Science and Engineering*, volume 3, pages 52–59, 2009. [cited at p. 12, 47, 48]

[93] L. Kissner and D. Song. Privacy-preserving set operations. *Advances in Cryptology - CRYPTO 2005*, 3621:241–257, 2005. [cited at p. 12, 47]

[94] N. Kiukkonen, J. Blom, O. Dousse, D. Gatica-Perez, and J. Laurila. Towards Rich Mobile Phone Datasets: Lausanne Data Collection Campaign. In *ACM International Conference on Pervasive Sciences*, 2010. [cited at p. 123]

[95] J. Krumm. A survey of computational location privacy. *Personal and Ubiquitous Computing*, 13(6):391–399, 2009. [cited at p. 53]

[96] P. Kumaraguru and L. F. Cranor. Privacy Indexes : A Survey of Westin's Studies, Institute for Software Research, Paper 856. Technical report, CMU, 2005. [cited at p. 135]

[97] Let's Meet There! http://www.letsmeetthere.net/, 2011. [cited at p. 52]

[98] J.R. Lewis. IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction*, 7, 1995. [cited at p. 39, 78]

[99] S.-D. Li and Y.-Q. Dai. Secure two-party computational geometry. *Journal of Computer Science and Technology*, 20, 2005. [cited at p. 82]

[100] J. Lin, G. Xiang, J.I. Hong, and N. Sadeh. Modeling People's Place Naming Preferences in Location Sharing. In *ACM International Conference on Ubiquitous Computing*, 2010. [cited at p. 136]

[101] Y.-H. Lin, A. Studer, H.-C. Hsiao, J. M. McCune, K.-H. Wang, M. Krohn, P.-L. Lin, A. Perrig, H.-M. Sun, and B.-Y. Yang. Spate: Small-group PKI-less authenticated trust establishment. In *International Conference on Mobile Systems, Applications, and Services*, pages 1–14, New York, NY, USA, 2009. ACM. [cited at p. 14, 54]

[102] K. Liu and E. Terzi. A Framework for Computing the Privacy Scores of Users in Online Social Networks. *ACM Transactions on Knowledge Discovery from Data*, 5(1):6, 2010. [cited at p. 147]

[103] F.M. Lord. *Applications of Item Response Theory to Practical Testing Problems*. ERIC, 1980. [cited at p. 147]

[104] C. Mancini, K. Thomas, Y. Rogers, B. A. Price, L. Jedrzejczyk, A. K. Bandara, A. N. Joinson, and B. Nuseibeh. From Spaces to Places: Emerging Contexts in Mobile Privacy. In *ACM International Conference on Ubiquitous Computing*, 2009. [cited at p. 145]

[105] S. Mardenfeld, D. Boston, S.J. Pan, Q. Jones, A. Iamntichi, and C. Borcea. Gdc: Group discovery using co-location traces. In *International Conference on Social Computing*, 2010. [cited at p. 2, 88]

[106] W. Mason and S. Suri. Conducting Behavioral Research on Amazons Mechanical Turk. *Behavior Research Methods*, 44(1):1–23, 2012. [cited at p. 137, 139]

[107] Y. Matsuo, N. Okazaki, K. Izumi, Y. Nakamura, T. Nishimura, and K. Hasida. Inferring Long-term User Property based on Users. In *International Joint Conference on Artificial Intelligence*, 2007. [cited at p. 2, 88, 106]

[108] MeetWays. http://www.meetways.com/, Visited Nov. 2011. [cited at p. 52]

[109] Mezzoman. http://www.mezzoman.com/, Visited Nov. 2011. [cited at p. 52]

[110] Microsoft Outlook. http://office.microsoft.com/outlook, Last visited 27.01.2011. [cited at p. 12, 46]

[111] Microsoft survey on LBS. http://go.microsoft.com/?linkid=9758039, 2011. [cited at p. 52, 53, 79]

[112] M. Miettinen and N. Asokan. Towards Security Policy Decisions Based on Context Profiling. In *ACM Workshop on Artificial Intelligence and Security*, 2010. [cited at p. 145]

[113] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. Campbell. Sensing Meets Mobile Social Networks: The Design, Implementation and Evaluation of the CenceMe Application. In *ACM Conference on Embedded Networked Sensor Systems*, 2008. [cited at p. 146]

[114] A. Narayanan and V. Shmatikov. De-anonymizing Social Networks. In *IEEE Security and Privacy*, 2009. [cited at p. 147]

[115] M.E.J. Newman. Fast Algorithm for Detecting Community Structure in Networks. *Physical Review E*, 69:5, 2004. [cited at p. 92]

[116] A Definitive List of NFC Phones. `http://www.nfcworld.com/nfc-phones-list/`, Last visited 13/Mar/2013. [cited at p. 106]

[117] T. Nicolai, E. Yoneki, N. Behrens, and H. Kenn. Exploring Social Context with the Wireless Rope. In *International Conference on On the Move to Meaningful Internet Systems*, 2006. [cited at p. 123]

[118] Nokia Ovi. http://ovi.nokia.com, Last visited 27.01.2011. [cited at p. 12]

[119] Nokia Qt framework. http://qt.nokia.com/, Last visited 27.01.2011. [cited at p. 33]

[120] A. Noulas, M. Musolesi, M. Pontil, and C. Mascolo. Inferring interests from mobility and social interactions. In *NIPS Workshop on Analyzing Networks and Learning with Graphs*, 2009. [cited at p. 2, 88]

[121] J.S. Olson, J. Grudin, and E. Horvitz. A Study of Preferences for Sharing and Privacy. In *Proc. of ACM CHI'05*, 2005. [cited at p. 135]

[122] F. Olumofin, P. Tysowski, I. Goldberg, and U. Hengartner. Achieving efficient query privacy for location based services. In *Privacy Enhancing Technologies Symposium*, 2010. [cited at p. 83]

[123] Orange Taxi sharing app, Visited Nov. 2011. [cited at p. 52]

[124] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. *Advances in Cryptology - EUROCRYPT '99*, 1592:223–238, 1999. [cited at p. 16, 26, 59, 60]

[125] G. Palla, AL Barabási, and T. Vicsek. Quantifying social group evolution. *Nature*, 446, 2007. [cited at p. 94, 97]

[126] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435, 2005. [cited at p. 92, 97, 109]

[127] S. Patil, Y. Gall, A. Lee, and A. Kapadia. My Privacy Policy: Exploring End-User Specification of Freeform Location Access Rules. In *Financial Cryptography and Data Security*, 2012. [cited at p. 139]

[128] S. Pidcock and U. Hengartner. Zerosquare: A privacy-friendly location hub for geosocial applications. In *Mobile Security Technologies*, 2013. [cited at p. 83]

[129] Please Rob Me. http://pleaserobme.com/, Visited Nov. 2011. [cited at p. 53]

[130] B. Priyantha, D. Lymberopoulos, and J. Liu. Littlerock: Enabling Energy-Efficient Continuous Sensing on Mobile Phones. *IEEE Pervasive Computing*, 10(2):12–15, 2011. [cited at p. 133]

[131] Qualcomm AllJoin. `https://developer.qualcomm.com/develop/mobile-technologies/peer-peer-alljoyn`, Last visited 13/Mar/2013. [cited at p. 106]

[132] S. Reeves. Internet is double-edged sword in arab revolts. http://middle-east-online.com/english/?id=46109, 2011. [cited at p. 88]

[133] O. Riva, C. Qin, K. Strauss, and D. Lymberopoulos. Progressive Authentication: Deciding When to Authenticate on Mobile Phones. In *USENIX Security*, 2012. [cited at p. 145]

[134] RL Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):126, 1978. [cited at p. 14, 54]

[135] MJB Robshaw and Y.L. Yin. Elliptic curve cryptosystems. *An RSA Laboratories Technical Note*, 1997. [cited at p. 60, 75]

[136] N. Sadeh, J. Hong, L. Cranor, I. Fette, P. Kelley, M. Prabaker, and J. Rao. Understanding and Capturing People's Privacy Policies in a Mobile Social Networking Application. *Personal and Ubiquitous Computing*, 13(6):401–412, 2009. [cited at p. 128, 145, 146]

[137] Samsung Galaxy S4. http://www.samsung.com/global/microsite/galaxys4/, Last visited 14/05/2013. [cited at p. 1]

[138] P. Santos and H. Vaughn. Where shall we meet? proposing optimal locations for meetings. In *Workshop on Map Based Interaction in Social Networks*, 2007. [cited at p. 82]

[139] B. Schoenmakers. A simple publicly verifiable secret sharing scheme and its application to electronic voting. In *International Conference on Cryptology*, pages 784–784. 1999. [cited at p. 74]

[140] B. Settles. Active learning literature survey. *University of Wisconsin, Madison*, 2010. [cited at p. 134]

[141] SFR Carte Wifi. `http://cartewifi.sfr.fr`, Last visited 13/Mar/2013. [cited at p. 122]

[142] M. C. Silaghi. Meeting scheduling guaranteeing n/2-privacy and resistant to statistical analysis (applicable to any discsp). In *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 711–715, Washington, DC, USA, 2004. IEEE Computer Society. [cited at p. 12, 47]

[143] M.C. Silaghi and D. Mitra. Distributed constraint satisfaction and optimization with privacy enforcement. *International Conference on Intelligent Agent Technology*, pages 531–535, 2004. [cited at p. 12, 35, 36, 47]

[144] I. Smith, S. Consolvo, A. Lamarca, J. Hightower, J. Scott, T. Sohn, J. Hughes, G. Iachello, and G. Abowd. Social Disclosure of Place: From Location Technology to Communication Practices. *Pervasive Computing*, pages 151–164, 2005. [cited at p. 128, 132, 145]

[145] A. Solanas and A. Martínez-Ballesté. Privacy protection in location-based services through a public-key privacy homomorphism. In *Public Key Infrastructure*, pages 362–368. 2007. [cited at p. 82]

[146] M. Stadler. Publicly verifiable secret sharing. In *Advances in Cryptology - EUROCRYPT '96*, pages 190–199, 1996. [cited at p. 45]

[147] J. C. Tang, N. Yankelovich, J. Begole, M. Van Kleek, F. Li, and J. Bhalodia. ConNexus to Awarenex: Extending Awareness to Mobile Users. In *ACM CHI Conference on Human Factors in Computing Systems*, 2001. [cited at p. 146]

[148] K. Tang, J. Hong, and D. Siewiorek. The Implications of Offering More Disclosure Choices for Social Location Sharing. In *ACM CHI Conference on Human Factors in Computing Systems*, 2012. [cited at p. 128]

[149] K. P. Tang, P. Keyani, J. Fogarty, and J. I. Hong. Putting People in Their Place: An Anonymous and Privacy-Sensitive Approach to Collecting Sensed Data in Location-Based Applications. In *ACM CHI Conference on Human Factors in Computing Systems*, 2006. [cited at p. 146]

[150] The Week Staff. The carrier iq cellphone scandal: 'an insane breach of trust'. The Week, December 2011. [cited at p. 3]

[151] S. Thurn and Y. I. Kane. Your apps are watching you. The Wall Street Journal, December 2012. [cited at p. 2]

[152] E. Toch, J. Cranshaw, P. H. Drielsma, J. Y. Tsai, P. G. Kelley, J. Springfield, L. Cranor, J. Hong, and N. Sadeh. Empirical Models of Privacy in Location Sharing. In *ACM International Conference on Ubiquitous computing*, 2010. [cited at p. 128, 129, 132, 134, 142, 145]

[153] E. Toch, J. Cranshaw, P. Hankes-Drielsma, J. Springfield, P. G. Kelley, L. Cranor, J. Hong, and N. Sadeh. Locaccino: A Privacy-centric Location Sharing Application. In *ACM International Conference on Ubiquitous computing (adjunct papers)*, 2010. [cited at p. 128]

[154] UTM coordinate system. https://www.e-education.psu.edu/natureofgeoinfo/c2_p21.html, Visited Nov. 2011. [cited at p. 54]

[155] R.J. Wallace and E.C. Freuder. Constraint-based reasoning and privacy/efficiency tradeoffs in multi-agent problem solving. *Artificial Intelligence*, 161(1-2):209–227, 2005. [cited at p. 12, 47, 48]

[156] J. Wiese, P.G. Kelley, L.F. Cranor, L. Dabbish, J.I. Hong, and J. Zimmerman. Are You Close with Me? Are You Nearby?: Investigating Social Groups, Closeness, and Willingness to Share. In *ACM International Conference on Ubiquitous computing*, 2011. [cited at p. 137, 140, 146]

[157] K. Xu, G.H. Yang, V.O.K. Li, and S.Y. Chan. Detecting dynamic communities in opportunistic networks. In *International Conference on Ubiquitous and Future Networks*, 2009. [cited at p. 94]

[158] M. Yokoo, K. Suzuki, and K. Hirayama. Secure distributed constraint satisfaction: Reaching agreement without revealing private information. *Artificial Intelligence*, 161(1-2):229 – 245, 2005. [cited at p. 12, 35, 36, 47, 48]

[159] E. Yoneki, D. Greenfield, and J. Crowcroft. Dynamics of Inter-Meeting Time in Human Contact Networks. In *International Conference on Advances in Social Network Analysis and Mining*, 2009. [cited at p. 111, 123]

[160] C. Zhang and Y. Huang. Cloaking locations for anonymous location based services. *Journal GeoInformatica*, 13(2):159–182, 2009. [cited at p. 83]

[161] D. Zhang, B. Guo, B. Li, and Z. Yu. Extracting social and community intelligence from digital footprints: An emerging research area. *Ubiquitous Intelligence and Computing*, 2010. [cited at p. 88]

[162] G. Zhong, I. Goldberg, and U. Hengartner. Louis, lester and pierre: Three protocols for location privacy. In *Privacy Enhancing Technologies*, pages 62–76. 2007. [cited at p. 82]

[163] A. Zunino and M. Campo. Chronos: A multi-agent system for distributed automatic meeting scheduling. *Expert Systems with Applications*, 36(3, Part 2):7011 – 7018, 2009. [cited at p. 48]

Index

Access control
    policies, 144
Active adversary, 72
Active attacks, 45, 51
Active learning, 127, 142
Amazon Mechanical Turk, 134
Android, 2
Asymptotic complexity, 34, 73

Behavior, 125
Bilinear mappings, 73
Boneh-Goh-Nissim encryption, 57, 59

Challenge-response
    game, 64
    methodology, 17, 63
Challenger, 18
Clustering algorithms, 90
Collusion, 72
Common availabilities, 11
Community statistics, 96
Confidence, 125, 139
    threshold, 132
Context, 1, 126, 143
Context-aware services, 85
Contributions, 4
Coordinate system, 52
Countermeasures, 119

Data-source diversity, 104
Decision-making, 144
Device-to-device, 104
Device-to-device communication, 86
Distributed constraint satisfaction, 47
Dummy questions, 137

E-voting, 47
ElGamal encryption, 22, 57
Encounter, 109

Extensions, 79

Features, 109, 130

Goldwasser-Micali encryption, 29, 47

Homogeineity tests, 138
Homomorphic properties, 13

Identifiability advantage, 17, 63
Inference, 103
Information request, 130
Information-sharing policies, 126
Intelligent agents, 47
Internet, 1
IP address, 129

Jaccard index, 97

k-center, 56

Linkability advantage, 17, 63
Location privacy, 101
Location-based services, 50

MAC address, 127
Machine learning, 5, 126
    classifier, 131
Meeting location determination, 5
    fair, 51
Meeting scheduling, 5, 11
    extensions, 43
    management applications, 49
    server, 15
Mobile social networks, 1

Network
    layers, 3
Nokia
    N810, 32

Nokia Instant Community, 88, 104

Oblivious, 56
Online survey, 134
Order-preserving, 62

Pailler encryption, 26
Performance, 73
    measurement, 34
Physical co-presence, 1
Privacy
    community, 87, 97
    dimensions, 3
    information, 2
    protection mechanisms, 3
    requirements, 63
Privacy attitude, 40
Privacy-preserving scheduling, 12
Private-set intersection, 47
Probabilistic advantage, 63

Qt, 32

SchedElg, 22
SchedPa, 26
Secret sharing, 47
Secure multi-party computation, 47
Semi-honest adversary, 14, 53
Sensors, 1
Server privacy, 66
Server-privacy, 19
Sharing behavior, 126
Social communities, 5, 86, 103
Social graph, 91
Social relationships, 103, 107, 138
    classifier, 114
SPISM, 5, 125
Statistics
    survey, 138
Supervised learning, 111

Transformation function, 16, 54

User burden, 126
User experience, 42
User privacy, 63
User study, 38, 77, 87, 105, 134
    participants, 39, 77
    questionnaire, 39
    results, 40
User-privacy, 17

Validation criteria, 137

Weight functions, 91

IGOR BILOGREVIC

EPFL-IC-LCA1
Station 14
1015 Lausanne
http://people.epfl.ch/igor.bilogrevic Email : igor.bilogrevic@epfl.ch
Tel.: 0041 21 693 66 21 igor.bilogrevic@gmail.com

EDUCATION

2009 – 2014 **Ph.D. in Communication Systems – On Multi-Dimensional Privacy in Context-Aware Mobile Networks, EPFL,** Switzerland. Thesis director: Prof. Jean-Pierre Hubaux.

2007 – 2009 **M.Sc. in Communication Systems with specialization in Wireless Systems** (GPA 5.79/6.0), **EPFL**, Switzerland.
Key courses: Advanced Digital Communications, Radio Communication, Wireless and Mobile Networks Architecture, Antennae and Radiation, Information Theory and Coding, Internet Security and Privacy, IT Security Engineering, Cryptography, Game Theory, Pattern Classification and Machine Learning, Project and Human Resources Management, Global Business Environment, Professional Communication.

2007 – 2008 **Erasmus exchange student at the Royal Institute of Technology (KTH)**, Stockholm, Sweden. Wireless Systems.

2004 – 2007 **B.Sc. in Communication Systems, EPFL**, Switzerland.

2000 – 2004 High school "Lugano 1" with specialization in Physics and Applied Mathematics, Lugano, Switzerland.

WORK EXPERIENCE

Jul. '13 – Oct. '13 **Research Intern, PARC (a Xerox Company)**, California, USA.
Data analytics and online privacy (details omitted due to Non-Disclosure Agreements).

Jun. '10 – Oct '12 **Research Collaborator**, **Nokia Research Center**, Lausanne. 50% - 70% of my research is funded entirely by Nokia. Topics include privacy-preserving applications and protocols on mobile devices, pervasive community privacy, in-device privacy control and decision-making on mobile devices (details omitted due to Non-Disclosure Agreements).

Sep. '09 – Feb. '14 **Project manager** for more than 20 Master, PhD and full-time intern projects, in collaboration with the Nokia Research Center, Lausanne.

Sep. '09 – Feb. '14 **Research Assistant** at the Laboratory for Computer communications and Applications (LCA1), EPFL. Research on security and privacy in mobile communications. UMTS, LTE cellular networks expertise.

Jun. – Jul '09 **Research Intern** at the Laboratory for Computer communications and Applications (LCA1), EPFL. Research on wireless security topics and their possible applications in mobile ad hoc networks. The resulting paper has been published at the WiOpt 2010 International Symposium, Avignon, France.

Jun. – Jul. '08 **Telecommunication Systems Engineer (Internship)**, Swisscube pico-satellite, Swiss Space Center, EPFL. Analysis, development and testing of the antennae and baseband communication subsystems of SwissCube, the first satellite entirely made by the Swiss universities. Link budget parameters optimization and study of trade-offs between modulation techniques, BER and power requirements.

2004 – present **IT consultant** and computer system assembler for small and home office environments.

PATENTS AND AWARDS

2013 **Igor Bilogrevic,** Julien Freudiger, Emiliano De Cristofaro, Ersin Uzun,
Lead inventor on 3 privacy-related patents filed by PARC (a Xerox Company) to the US Patent office. The content is confidential.

2012 **Igor Bilogrevic**, Kévin Huguenin
Patent application filed by Nokia to the US Patent office. The content is confidential.

2012 Nokia: Imad Aad
EPFL: **Igor Bilogrevic**, Kévin Huguenin
Patent application filed by Nokia to the US Patent office. The content is confidential.

2012 EPFL IC School Teaching Award 2012

2011 Nokia: Imad Aad, Pentti Valtteri Niemi, Anthony Durussel, Emiliano De Cristofaro
EPFL: **Igor Bilogrevic**, Murtuza Jadliwala, and Jean-Pierre Hubaux, "Method and Apparatus for Preserving Privacy for Appointment Scheduling", Patent application filed by Nokia to the US Patent Office.

PROFESSIONAL SERVICES

2013 Editor of the book "Networks – Emerging Topics in Computer Science", iConcept Press Ltd., Australia.

2010 – present Reviewer for Journals: IEEE Transactions on {Mobile Computing, Vehicular Technology, Information Forensics and Security, Computing}, IEEE Communications Surveys & Tutorials, Springer Information Systems Frontier, IET Information Security.
Reviewer for Conferences: ACM CCS '11-'12, PETS '11-'12-'13, ICCCN '11-'12, IEEE MOST '12, IEEE IOFC '11, IEEE CNS '13, IEEE IPCCC '11-'12, IEEE PerCom '13, Pervasive '12, UbiComp '13, Financial

Cryptography (FC) 2011, HotWiSec '11.

PUBLICATIONS

- **I. Bilogrevic**, M. Jadliwala, V. Joneja, K. Kalkan, J.-P. Hubaux, I. Aad, Privacy-Preserving Location Sharing to Determine Optimal Meeting Locations on Mobile Devices, (under submission) IEEE Transactions on Information Forensics and Security.
- **I. Bilogrevic**, K. Huguenin, M. Jadliwala, F. Lopez and J.-P. Hubaux, P. Ginzboorg and V. Niemi, Inferring Social Ties in Academic Networks Using Short-Range Wireless Communications. Workshop on Privacy in the Electronic Society (WPES), 2013.
- **I. Bilogrevic**, K. Huguenin, B. Agir, M. Jadliwala and J.-P. Hubaux. Adaptive Information-Sharing for Privacy-Aware Mobile Social Networks. 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2013), Zurich, Switzerland, September 8-12, 2013.
- **I. Bilogrevic**, K. Huguenin, M. Jadliwala, F. Lopez and J.-P. Hubaux, P. Ginzboorg and V. Niemi, Inferring Social Ties in Pervasive Networks: An On-Campus Comparative Study. ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2013), 4-page short paper, Zurich, Switzerland, September 8-12, 2013.
- M. Jadliwala, **I. Bilogrevic** and J.-P. Hubaux. Optimizing Mix-zone Coverage in Pervasive Wireless Networks, accepted in Journal of Computer Security (JCS), 2013.
- **I. Bilogrevic**, M. Jadliwala, I. Lam, I. Aad and P. Ginzboorg, V. Niemi, L. Bindschaedler and J.-P. Hubaux, Big Brother Knows Your Friends: on Privacy of Social Communities in Pervasive Networks. 10th International Conference on Pervasive Computing, Newcastle, UK, June 18-22, 2012.
- L. Bindschaedler*, M. Jadliwala*, **I. Bilogrevic**, I. Aad and P. Ginzboorg, V. Niemi and J.-P. Hubaux, Track Me If You Can: On the Effectiveness of Context-based Identifier Changes in Deployed Mobile Networks. 19th Annual Network & Distributed System Security Symposium (NDSS), San Diego, California, USA, February 5-8, 2012.
- M. Jadliwala, **I. Bilogrevic** and J.-P. Hubaux. Optimizing Mixing in Pervasive Networks: A Graph-Theoretic Perspective. 16th European Symposium on Research in Computer Security (ESORICS), Leuven, Belgium, September 12-14, 2011.
- **I. Bilogrevic**, M. Jadliwala, P. Kumar, S. S. Walia and J.-P. Hubaux, I. Aad and V. Niemi, Meetings through the Cloud: Privacy-Preserving Scheduling on Mobile Devices, in Journal of Systems and Software, Special Issue on 'Mobile Applications: Status and Trends', vol. 84, p. 1910-1927, 2011.
- **I. Bilogrevic**, M. Jadliwala, K. Kalkan, J.-P. Hubaux and I. Aad. Privacy in Mobile Computing for Location-Sharing-Based Services . The 11th Privacy Enhancing Technologies Symposium (PETS), Waterloo, Canada, July 27-29, 2011.
- **I. Bilogrevic**, M. Manshaei, M. Raya and J.-P. Hubaux. OREN: Optimal Revocations in Ephemeral Networks, in Computer Networks, vol. 55, p. 1168-1180, 2011.
- **I. Bilogrevic**, M. Jadliwala, J.-P. Hubaux, I. Aad and V. Niemi. Privacy-Preserving Activity Scheduling on Mobile Devices. First ACM Conference on Data and Application Security and Privacy (CODASPY), San Antonio, Texas, USA, February 21-23, 2011.
- **I. Bilogrevic**, M. Jadliwala and J.-P. Hubaux. Security Issues in Next Generation Mobile Networks: LTE and Femtocells. 2nd International Femtocell Workshop, Luton, UK, June 21, 2010.
- **I. Bilogrevic**, M. Manshaei, M. Raya and J.-P. Hubaux. Optimal Revocations in Ephemeral Networks: A Game-Theoretic Framework. 8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt 2010), Avignon, France, June 1-3, 2010.

TEACHING

Teaching assistantship	**Teaching Assistant** for the courses: Mobile Networks, Security and Cooperation in Wireless Networks, Information Technologies Project, Computer Science I. Managing and supervising more than 300 students and more than 20 student assistants.
	Student assistant for the courses: "Principles of Digital Communications" taught by Prof. Rimoldi, "Circuits and Systems I+II", taught by Prof. Hasler, EPFL.
Full-time interns (3 months)	(at **EPFL**)
2013	Omar Bougamza, "Smart Privay-Aware Information-Sharing Mechanism"
	(at **Nokia Research Center**)
2012	Ayush Patwari, Shrey Sharma, "Location Privacy Visualization"
2011	Istvan Lam, "Social Community Privacy", Istvan is the CEO of Tresorit, a cloud-based secure computing platform
2011	Saurav Bhattacharya, Saurabh Goyal, "Privacy of Social Relationships and Communities"
2010	Sudeep Singh Walia and Praveen Kumar, "Privacy-preserving scheduling", Sudeep is now with Amazon and Praveen is with IBM Research
2009	Tali Gutman, "Privacy Visualization", Tali is now with Google
Master Thesis	
2011	Bindschaedler Laurent, "Nokia Instant Community Security and Privacy"
Master Research	
2013	Stefan Mihaila, "Understanding the motivation behind location-related (semantic) check-ins and the utility implications"

PhD Projects

2013 Shahid Nauman, “Understanding the motivation behind location-related (semantic) check-ins and the utility implications”

2013 Anh Pham Thi Van, “Secure contextual evidences”

2010 Bongard Dominique, “Catch me if you can: UMTS and Femtocell Security and Privacy”

Semester projects

2013 Ilia Kebets, Claire Musso, “Smart Privay-Aware Information-Sharing Mechanism”

2012 Hamza Harkous, “Extracting the Most Influential Factors in Privacy Decisions”

2012 Tang Xin, Peng Gao, Aylin Nerrah, “Smart Privay-Aware Information-Sharing Mechanism”

2011 Natalyia Arbusova, “QR-code Security”

2011 Juyuan Liu, “Familiar Stranger Applications for Pervasive Social Networks”

2011 Vasileios Agrafiotis, “Private Information Exposure in Online Social Networks with iOS, Android and Symbian Mobile Devices”

2011 Florent Lopez, Michalko Matej, “Social Community Privacy”

2010 Carl Hedari, Charles Edmond Renouard, “UMTS Femtocell Security and Privacy”

2010 Vishal Joneja, “Optimal Meeting Location Determination”

ADDITIONAL INFORMATION

Activities — Member of the ACM, the IEEE and the Swiss Informatics Society (SIS).
Former member of the AIESEC student organization. AIESEC helps students to develop their professional and human skills.

Hobbies — Kung Fu martial art, windsurf, golf, reading (fiction, thriller), cinema, running, fitness, hiking.

PERSONAL SKILLS AND COMPETENCES

Languages — Italian & Croatian as mother tongues, English & French fluent written and spoken, German satisfactory written and spoken, Swedish elementary.

Social skills — Team worker with focus on development, human relations and excellence. Used to talk different languages and dealing with different cultures. Innovator and initiator.

Organizational skills — Competency, responsibility, flexibility and adaptability. Strong analytical and problem solving skills.

Technical skills — Java, C/C++, XML, MySQL, Matlab, R, HTML programming skills, Ansoft HFSS radio engineering software, Ericsson TEMS software for cellular networks deployment, MS Office, Windows, Linux.

ACADEMIC PROJECTS

Feb. ‘12 – Jun. ‘12 **PhD project, “Was Orwell Right? On Privacy of Social Relationships and Communities in Ubiquitous Networks”** (Grade 6/6). The resulting paper was published at the Workshop on Privacy in the Electronic Society (WPES 2013), Berlin, Germany.

Feb. ‘11 – Jun. ‘11 **PhD project, “Where do we Meet? A Privacy-Preserving Meeting Location System for Mobile Devices”** (Grade 6/6). The resulting paper was published at the 11th Privacy Enhancing Technologies Symposium (PETS 2011), Waterloo, Canada.

Sep. ‘09 – Jan. ‘10 **PhD project, “Security and Privacy in Next Generation Mobile Networks: LTE and Femtocells”** (Grade 6/6). The resulting paper was published at the 2nd International Femtocell Workshop (2010), Luton, UK.

Feb. – Jun. ‘09 **M.Sc. Thesis “Revocation Games in Ephemeral Networks”** (Grade 6/6) **at LCA1, EPFL.** Thesis focused on design, implementation and performance evaluation of an efficient security scheme for mobile ad hoc and pervasive networks.

Jan. – Jun. ‘08 **Network System Designer** for the Communication Systems Design project Open Source in African Networks (OSiAN, semester project, grade 6/6)**,** Telecommunication Systems Lab, KTH, Stockholm, Sweden.

Mar. – Jun. ‘08 **Research paper** “VoIP over 802.11 WLANs: Enabling Qos, High Capacity and Fast Handoffs in a Wireless Environment” , Practical VoIP course (KTH).

Jan. – Mar. ‘08 **Research paper** “Satellite Communications: Internet challenges and strategies for low-latency communications” (Grade 6/6), Wireless and Mobile Network Architectures course (KTH).

Mar. – Jul. ‘07 **SwissCube Telecommunication Systems** (semester project, grade 6/6), Swiss Space Center, EPFL.