DISS. ETH NO. 22772
DISS. TIK NO. 155

Privacy-Preserving Use of Social Information in Opportunistic Networks

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH Zurich

(Dr. sc. ETH Zurich)

presented by

BERNHARD DISTL

Master of Science, ETH
born March 20, 1980
citizen of Austria

accepted on the recommendation of
Prof. Dr. Bernhard Plattner, examiner
Prof. Dr. Gunnar Karlsson, co-examiner
Dr. Franck Legendre, co-examiner

2015

Acknowledgments

I am deeply grateful that Prof. Bernhard Plattner gave me the opportunity to perform my thesis in his group. He gave me guidance throughout the work and also supported me during difficult times I had to go through in my private life. He also backed me up and gave me confidence during my time at Uepaa, where business and research interests were not always in line with each other.

I thank Franck Legendre and Theus Hossmann, who advised my over the course of my work. They challenged my ideas, helped me refine them and put them down on paper. I also thank Prof. Gunnar Karlsson for the discussions and feedback throughout my work, especially for his offer to be co-examiner for this thesis.

It was a great pleasure to work with my colleagues at the communication systems group. I specially thank my office mates Sascha Trifunovic, Ehud Ben-Porat and Elias Raftopoulos for the great atmosphere in our office that combined hard work and fun perfectly.

Prof. Plattner and Franck Legendre also gave me the opportunity to conduct a part of my Ph.D. work in cooperation with Uepaa AG. I learned a lot during this time and also enjoyed to put research into practice. My thanks extend to the whole team at Uepaa: David Altorfer, Björn Muntwyler, Tal Heskia, Xaver Weibel, Dominic Bestler and Mathias Haussmann.

The support of my parents during all the years was essential that I could pursue my studies and Ph.D. work.

The loving support of Regina kept me going and I can not count all the things she did for me, so that I could focus on my work. I am immensely grateful for her being my partner, motivating me when I needed it and replenish my energy with all the beautiful things we experienced and shared in the past years.

Abstract

The popularity of smartphones and other mobile devices has lead to a significant increase in the use of cellular mobile data. As this demand is predicted to continue to grow exponentially in the future, providing sufficient and ubiquitous cellular coverage becomes increasingly challenging and sometimes even unfeasible. In addition to the generally increasing demand, unpredictable crowds are even more challenging to accommodate with existing infrastructure. Even worse, natural or man-made disasters may break infrastructure, thus disrupting connectivity. Also, a lack of economic incentives prevents remote and rural regions from receiving sufficient connectivity.

Opportunistic networks are envisioned to mitigate many of those issues. Mobile devices can leverage wireless capabilities (e.g. Bluetooth, WiFi adhoc, WiFi direct) to communicate directly with each other, whenever two devices are in mutual transmission range (in *contact*). State of the art opportunistic networks extract a stable structure (a *contact graph*) from all available contacts. The contact graph can then be used to enable multi-hop communication to maintain connectivity and provide services in an opportunistic manner.

Within the context of opportunistic networks, there is a close relationship between a device and its user. Thus, all communication of one device can be attributed to its user. This highly personal relationship creates severe privacy issues. What content the user consumes, which other users he or she meets and where the user is located can all be easily determined by an interested party. While some privacy aspects (e.g. location) are well investigated, the privacy of social information (e.g. friendships, social structure) has not received much attention. With the first two of the three contributions of this thesis, we investigate the use of social information in the two fundamental building blocks of opportunistic networking: routing and neighbor detection. We find that state of the art solutions do not protect privacy relevant infor-

mation. Therefore, we provide contributions in the area of privacy preserving use of social information within opportunistic networking.

In the *first* contribution, we present an algorithm to protect the privacy sensitive information associated with the social structure encoded by a contact graph as it is used by state of the art routing algorithms for opportunistic networks. A first straightforward approach to hide the social structure by randomly modifying the contact graph quickly leads to wrong routing decisions and thus diminishes opportunistic networking performance. By changing edges in the graph more selectively, we can maintain essential features of the graph that are required for routing (e.g. centrality ranking) while still effectively hiding the social structure encoded in the edges of the graph. We design a step-wise optimal greedy algorithm and a heuristic variant that can be calculated faster and for larger graphs. Eventually, we evaluate routing performance by using privacy protected graphs in existing state-of-the-art opportunistic routing algorithms.

In an anonymous opportunistic network, even friends no longer can recognize each other. Thus, they no longer can rely on existing (real world) social links (e.g. friendships) for security and trust mechanisms. In the *second* contribution, we present a protocol to detect once established social links in a privacy protected manner while maintaining the anonymity provided by a perfectly anonymous opportunistic network. Our hash based protocol can recognize pre-established social links among nodes *without* revealing private information, hence protecting users identity and social information. We implement our algorithm in a smartphone application and evaluate its performance.

Complementary to protecting the privacy of social information in the two fundamental building blocks of opportunistic networks (routing and neighbor detection), our *third* contribution investigates how current generation smart phones impact the physical contact events and thus contact modeling. The radio properties of smartphones (e.g. transmission range, directionality) are an important factor that shape contact events between two devices. However, in contrast to user mobility, which is well investigated, the impact of radio properties of smartphones on wireless contacts has to our knowledge not been investigated so far. We hence investigate the WiFi radio performance of smartphones for opportunistic networking. We start by revisiting the classical link budget, later adding the impact of the phone's carrier. We then perform extensive measurements to fully characterize all components of the link budget between two smartphones. Our measurements also give a clear indication of which of the existing propagation models is suited best for smartphones in

a pedestrian outdoor setting. Finally, to assess the capacity of opportunistic networking, we evaluate a simple scenario of two pedestrians crossing on a path.

Kurzfassung

Durch die steigende Verbreitung und Populartiät von Smartphones hat sich die Internetnutzung stark verändert. Dank der Mobilfunknetzte konsumieren Menschen Internetinhalte nicht mehr nur zu Hause oder am Arbeitsplatz sondern praktisch überall. Der stetige Ausbau der Mobilfunknetze hat auch dazu geführt, das die Internetabdeckung heute bisher ungekannte Ausmasse erreicht hat. Das Internet der Dinge wird den Druck die Mobilfunknetze auszubauen noch verstärken. Allerdings werden durch diese Ausdehnung die Grenzen der Internetverfügbarkeit unschärfer und die Belastung durch steigende Datenvolumen immer größer.

Opportunistische Netzwerke nutzen direkte, drahtlose Kommunikation von Gerät zu Gerät um Informationen zwischen den Benutzern auszutauschen. Dank drahtloser Kommunikationstechnologien wie WLAN oder Bluetooth können von Smartphones direkt miteinander kommunizieren ohne auf Infrastruktur angewiesen zu sein. Wenn die herkömmlichen Mobilfunknetze überlastet oder nicht verfügbar sind, eröffnet opportunistische Kommunikation neue Möglichkeiten für Anwender. Informationen werden immer dann ausgetauscht, wenn zwei Benutzer gegenseitig in Funkreichweite kommen. Zusätzlich tragen Benutzer die Daten auch physikalisch mit sich herum wenn sich in ihrem Alltag bewegen. Obwohl theoretisch genug Potential besteht, haben sich opportunistische Netzwerke wegen fehlender Geschäftsmodelle bis jetzt noch nicht durchgesetzt. Wir untersuchen drei Themen von denen wir glauben, daß sie die praktische Akzeptanz von opportunistischen Netzwerkern erleichtern werden. Als erstes behandeln wir den Schutz der sozialen Informationen die für Routing genutzt werden, als zweites machen wir soziale Kontakte in einem anonymen Umfeld nutzbar und als drittes evaluieren wir Smarphone WLAN Eigenschaften um besser zu verstehen wie Kontakte durch die eingesetzte Technolgie beeinflusst werden.

Zuerst behandeln wir soziale Informationen, die ein wichtiges Werkzeug sind um opportunistische Netzwerke zu verbessern. Unter anderem kann man mit einem Kontaktgraphen, der soziale Strukturen erfasst, die Erfolgsrate bei Routing maximieren. Dabei untersuchen wir das Verhältnis von Schutz der Privatsphäre und Routingerfolg im Kontaktgraphen. Durch kontrolliertes hinzufügen und entfernen von Kanten können wir den Schutz der echten sozialen Verbindungen verbessern. Anhand von künstlichen und realen Kontaktgraphen zeigen wir erst, dass zufällige Änderungen am Graphen das Ranking von Betweenness centrality (einer typischen Routingmetrik) schnell zerstört wird und zu falschen Routingentscheiden führt. Unser Ansatz immer die Kante mit dem kleinsten Einfluss auf das Ranking zu ersetzen führt dazu, dass die Nutzbarkeit erhalten bleibt während gleichzeitig ein hohes Schutzniveau erreicht wird. Die Skalierbarkeit unserer Lösung wird durch eine Heuristik basierend auf der Ähnlichkeit von Knoten erreicht. Sie modelliert den Basisalgorithmus und ist für grosse Graphen gut einsetzbar. Daß Routing auch mit den geschützten Graphen funktioniert können wir zeigen, in dem geschützte Graphen für routing mit einem bekannten Algorithmus eingesetzt werden.

Im zweiten Teil bewegen wir uns im Spannungsfeld zwischen Anonymität und der Nutzbarkeit von sozialen Informationen. Einerseits müssen soziale Informationen geschützt werden, andererseits können Sie genutzt werden um die Leitung und Sicherheit in opportunistischen Netzwerken zu steigern. Vollständig anonyme opportunistische Netze verhindern jedoch das Wiedererkennen von bestehenden sozialen Verbindungen (z.B. Freundschaften). Wir entwerfen ein Protokoll, dass einen Ausweg aus diesem Dilemma darstellt. Es ermöglicht, einmal erstellte soziale Verbindungen zwischen Benutzern wiederzuerkennen, *ohne* daß private Informationen einsehbar werden, wodurch es möglich wird, Anonymität *und* Leistung und Sicherheit zu unterstützen. Unser Protokoll nutzt mit Bloomfiltern eine Hash-basierte Konstruktion die gleichzeitig dem Schutz der sozialen Informationen und einer zuverlässigen und schnellen Erkennung bestehender sozialer Verbindungen in einem anonymen Umfeld dient. Zusätzlich implementieren und evaluieren wir das Protokoll auf Android Smartphones.

Zuletzt wenden wir uns der Tatsache zu, dass (Funk)Kontakte bisher vor allem vom Standpunkt der Benutzermobilität aus untersucht wurden. Allerdings hängen die Eigenschaften der Kontake zu einem großen Teil auch von den involvierten Geräten (meistens Smartphones) ab. Unseres Wissens nach hat sich bisher noch keine Studie mit dem Einfluss der Funkeigenschaften moderner Smartphones auf opportunistische Kontake befasst. Darum unter-

suchen wir in diesem Teil der Arbeit die Auswirkungen der WLAN Charakteristik von Smartphones auf opportunistische Kommunikation. Wir beginnen mit dem klassischen Linkbudget und erweitern diese Betrachtung um den Einfluss des Menschen der das Gerät trägt. Des Weiteren führen wir Messungen durch, die es uns erlauben alle Komponenten des Linkbudgets zwischen zwei Smartphones zu charakterisieren. Das zwei Strahlen Ausbreitungsmodell stellt sich als das beste Modell für Smartphones heraus. Zuletzt betrachten wir ein Szenario in dem sich zwei Fussgänger begegnen und berechnen die Datenübertragungskapazität der daraus resultierenden opportunistischen Kontakts.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the past years, smart mobile devices and data services have changed the way people live and organize their lives. Always available information and communication possibilities reshape many aspects of life, such as how personal relationships develop, how political and social movements are organized and how businesses offer their services. Today, smartphones are omnipresent. In 2014, for the first time, the number of smartphones sold exceeded the 1 billion mark according to a market analysis by Gartner [37]. In addition to the increasing popularity of smartphones, the number of downloaded applications on smartphones accumulated to approximately 100 billion in 2013 while in 2015 this number was already reached by the Apple App Store alone [88, 89]. This huge success drives the exponential growth for mobile data [18] which is expected to exceed 24 exabytes per month in 2019. On the other hand, mobile networking technologies based either on cellular networks (3G/4G/LTE) or on IEEE 802.11 WiFi technology have to satisfy the data communication needs of smartphones and their applications. However, those technologies have difficulties keeping up with the increasing demand for mobile data in terms of coverage and capacity. Especially during peak periods and events that attract many people, it is difficult or even unfeasible to provide enough bandwidth with existing cellular infrastructure networks. Additionally, failure of the communication infrastructure, as it can be caused by natural or man-made disasters, leads to a complete communication breakdown. Finally, it sometimes is economically not feasible to deploy expensive cellular network infrastructure in sparsely populated regions.

For the foreseeable future, the areas that are uncovered by conventional infrastructure will remain. Therefore, other communication paradigms can provide means to fill the gaps. One possible solution is to utilize direct device to device communication that does not require any infrastructure to transfer data. Smartphones today, as well as other mobile devices, support a variety of wireless standards with device to device communication capabilities such as Bluetooth [12], WiFi Direct [104] or LTE Direct [80]. Utilizing those technologies activates an additional potential for communication independent of the range of centralized infrastructure such as cellular base stations or WiFi access points. Whenever two devices come into mutual transmission range (in *contact*), they can use this contact to exchange data. The range within such a contact can occur depends on the wireless technology, as well as the environment (e.g. indoor or outdoor) and can range anywhere from a few meters to a few hundred meters.

Using a single contact to exchange data is simple, but providing services where contacts are unpredictable and numerous is very challenging. Opportunistic networks are designed to provide services in such challenging environments. Social information is a major factor that state of the art solutions for opportunistic networks use to overcome the challenges. The social behavior of users in opportunistic networks creates structure in contacts when users follow their daily routines and they trust information from their friends more than from strangers.

Social information is very closely linked to individuals and thus a privacy sensitive topic. Especially in the post Snowden aera [1] privacy has become a major topic of concern for the general public. In order to gain acceptance of a broad audience for new electronic communication technologies such as opportunistic networks, privacy needs to be protected by design. While privacy in general has a very broad scope, in this thesis we will focus on essential features in opportunistic networks that make use of privacy sensitive information. It turns out, that for essential features (like routing or neighbor detection, see Section 1.4) social information plays an important role in state of the art solutions.

With social information being used and therefore readily available for everybody, a number of questions about the sensitivity and use of social information arise, such as *What kind of social information is used in opportunistic*

[1] Edward Snowden revealed documents that show how intrusive and powerful NSA methods are and to what extent it collects and analyzes privacy sensitive information. The material has been reviewed and partially released in cooperation with Glen Greenwald and The Guardian newspaper `http://www.theguardian.com/us-news/edward-snowden`

networks? How is social information used in opportunistic networks? Can available social information be misused? Can the social information be protected? Does the protection of social information impact the performances of opportunistic protocols or networks? In this dissertation we strive for answers to some of those questions. In order to protect social information, we need to understand what kind of social information is used in opportunistic networks today, for which functions of the network it is used and whether or not all of the currently collected information is critical to the operation of the network.

We focus on two fundamental building blocks of opportunistic networks (neighbor detection and routing) which are present in all implementations. Solutions for those building blocks can be integrated into every opportunistic network with minimal effect on the application running on top of it. The challenges introduced by the focus on the fundamental building blocks routing and neighbor detection are

- maintaining routing performance while minimizing or removing social information that is visible to attackers and doing so in a centralized as well as distributed manner.
- to detect existing social links among two nodes in contact without revealing the users identities or friendships to anybody else and doing so in a fast and efficient way to preserve the scarce contact time for data exchange.

The first contribution of this thesis addresses the privacy of social information that is used by routing in opportunistic networks. We find, that state of the art routing protocols base their decisions on a graph that encodes the social structure of the users in its edges (see Section 1.4.2). Metrics from complex networks analysis (e.g. centrality) are calculated on the graph and then used to make forwarding decisions. A first approach to randomly modify the graph fails to maintain correct routing decisions. Thus, we design an algorithm that protects the social information encoded in the edges of the graph while at the same time it maintains correct routing decisions in the network.

In the second contribution, we look at how social information is used in neighbor detection of opportunistic networks. Connections to social links are used to improve various communication aspects such as trusted sources of information or spam prevention. In order to support those aspects without sacrificing users privacy, we design a fast and efficient protocol to detect existing social links without revealing the users identities or information about

their social links. This enables users to benefit from their social links even in a situation where they want to otherwise act anonymously.

A central aspect of opportunistic networking are the contacts between users smartphones. The third contribution of this thesis investigates how current smartphone design impacts the opportunistic contacts. While mobility of users is well investigated, the radio characteristics of smartphones, which is the second important factor for when a contact occurs, has not received much attention. Based on the link budget between two smartphones, we characterize the individual elements of the link using multiple information sources such as publicly available specifications and our own measurements. Based on our measurements, we are also able to select the best matching available propagation model for smartphone outdoor use. Finally, we use our results to estimate the data transfer capacity of a contact between two pedestrians crossing on a street.

The remaining sections of this introduction provide motivation for opportunistic networking in general in Section 1.1 and describe existing architectures and their commonalities in Sections 1.2 and 1.3. In Section 1.4 we narrow the focus to how social information is used in opportunistic networks and argue for the need for privacy in Section 1.5, before we specifically review related work on the use of social information and privacy in Section 1.6. The introduction concludes with the contributions made by this thesis and an overview over the remaining chapters in Section 1.7.

1.1 Opportunistic network scenarios

The introduction of smartphones in the global market has changed how people perceive and use mobile communication. Within a few years, smartphones transformed from a luxury gadget for rich into an every day companion for everybody. The recent development of low cost smartphones has further pushed smartphone adoption across the globe, especially in emerging countries such as India, Russia and Mexico. Market research by Gartner [37] estimates the number of smartphones sold in 2014 alone to exceed 1.24 billion devices (up from 923 million in 2013). The evolving feature set and computing power of smartphones also change how people access Internet services. Already for 2015, an estimated 788 million people will access the Internet via mobile devices only, without using any traditional fixed computing infrastructure at all [90].

A consequence of the increasing popularity and versatility of smartphones

is the rapidly rising demand for data over cellular networks. A study by Cisco Inc. [18] indicates, that in the end of 2014 global mobile data traffic was approximately 2.5 Exabyte per month, an increase of 69 percent from 2013. Specifically for smartphones, it estimates that data usage per individual smartphone has increased by 45 percent last year from 563 MB per month in 2013 to 819 MB in 2014. The numbers are evidence for a worldwide adoption of smartphones, which has led to the availability of information almost any time and at any place. As smartphones connected to the mobile Internet provide a vast amount of services, asking "*why should smartphones connect directly to each other?*" is a valid question. The growth in data demand, as well as the dependence of users on their smartphones, comes with its own challenges. In situations where traditional cellular infrastructure struggle, fail or simply can not deliver a sought after service, opportunistic networks may provide what otherwise would not be available. The following subsections outline some scenarios, where opportunistic networking provides benefits for users. Such opportunistic networking use-cases can be separated into four categories: (a) cellular data offloading, which may prevent overloading the mobile Internet infrastructure; (b) communication where cellular infrastructure has broken down or is completely missing; (c) providing an additional or alternative communication channel for security purposes, as opportunistic networking can provide a means of communication that is much more difficult to intercept; (d) new applications, that become possible by opportunistic communication, such as sharing information in a bounded area.

1.1.1 Cellular data offloading

The immense popularity of smartphones is driven by their increasing versatility which facilitates many aspects of peoples everyday lives. This versatility of smartphones is caused by the staggering and ever increasing number of applications available [88] and provides more and more functionality for users in every aspect of their lives. The main enablers for the variety of applications are continuously growing computing power and connection speed of smartphones. Those two factors are essentially driving the demand for data, as higher data volumes can be retrieved and processed within a shorter period of time. Additionally, data intensive tasks such as video streaming account for a growing share of mobile data traffic, which by 2014 represented approximately 55 percent [18] of the global mobile data traffic volume. Also, high connection speeds allow to outsource computing intensive tasks to the cloud (e.g. virtual personal assistants such as Apple Siri, Google Now or

Microsoft Cortana). Finally, a significant portion of mobile Internet traffic is already offloaded to the fixed network via 802.11 wireless networks. According to Cisco [18], 46 percent of the global mobile Internet traffic was offloaded from the cellular infrastructure through WiFi networks and femto cells in 2014. Without this effect, mobile data would have effectively grown 84 percent in 2014 instead of 69 percent.

Even with high investments in the networks, deploying new cellular technologies such as LTE [28] and the use of femto cells, there are situations where the cellular infrastructure can not cope with the demand. Especially during events with many people and other unpredictable crowds, demand peaks exceed the available capacity by far. Those peaks can only partially be anticipated, therefore installing additional infrastructure such as femto cells is costly and sometimes even unfeasible. Offloading traffic by means of opportunistic networking [44, 62] can help handling peaks in demand. Offloaded data is restricted to delay tolerant applications and best suited when multiple users are interested in the data at the same geographic location. Offloading reduces the required infrastructure bandwidth by downloading a piece of content once over the cellular infrastructure and then propagating it from device to device using opportunistic networking. Thus, each additional user that is interested in the content does not need to download it via its cellular connection any more.

1.1.2 Communication in the absence of infrastructure

Today, many people are used to being connected at all times. However, being connected in this way is more a privilege than anything else, as the digital divide [42] separates connected and unconnected areas of the world. Even worse, the connected part of the world may experience disconnection events caused by natural or man made disasters, which bring down the established infrastructure. Bridging the digital divide by using device to device communication has been the focus of various initiatives [29, 43, 91].

In contrast to regions with limited or no coverage beyond the digital divide, situations where natural or man made disasters cause an infrastructure break down can not be limited to specific geographic areas. Furthermore, the frequency of such events, already on a European scale with 137 events in 2014 [17], is higher than one might expect. One prominent example is the earthquake that caused a tsunami in Japan in 2011. In this event, communication infrastructure was shut down or completely destroyed. Several

regions remained without connectivity for several days [36]. In such a situation, communication is essential for first responders and people in the area to quickly act and react properly. Opportunistic networks can help establish and maintain communication in such a setting and thus aid in the efforts of rescue services to bring help to the right places and people faster. Additionally, they can allow people to organize themselves to help others, find missing relatives or find safe escape routes. In such a case, information is transmitted from device to device until it reaches one with cellular connectivity, from where it can be sent on to the fixed infrastructure [47].

1.1.3 Alternative communication channel

Data transmitted over the Internet passes through numerous networks which are controlled by different providers and countries. As such, data transmitted is subject to commercial, criminal and national interests. Large Internet companies (e.g. Yahoo, Google, Microsoft, ...) collect and analyze data to increase their revenues [38], criminal efforts may eavesdrop, block, alter or fake legitimate data traffic for its purposes [99] and national actors may do the same for their own motivation (e.g. large scale surveillance [41], industrial espionage or censorship). Opportunistic networks offer alternative communication channels that are out of reach of "traditional" Internet surveillance. They can be used instead of, or additionally to the Internet and provide increased privacy and security for the users. For example, users can distribute and access content or opinions that would otherwise be censored via an opportunistic network. Nevertheless, opportunistic networks are not immune to attacks themselves. However, those attacks can not be done with existing Internet attack tools and typically incur a higher cost for the attacker due to the local nature of opportunistic networks. Communication patterns and location of users can still be tracked [85] from which locations of interest, such as work or home, or social structures [26] can be derived.

1.1.4 New services

The properties of opportunistic network design, which is based on direct local contacts also opens room for new services that exploit the local properties of opportunistic networks. Two examples are geo-local (or "floating") information and local social networks.

geo-local information Street map tiles and temporary local advertisements

are just two examples of information that has only limited geographic utility. Such services, where information is only usable in a small local area, can be provided by opportunistic networks. In contrast to existing location based services, that determine the location via GPS and then download relevant information from an Internet server, opportunistic content can float [53] around in an area readily available for everybody around. The floating content does neither depend on a GPS (satellite) based location, nor on an available Internet connection. This makes it suitable for indoor and underground settings as well as touristic places, where roaming fees for data connections can incur significant charges for tourists.

local social networks A large part of peoples lives is based on local interactions. Being with family, co-workers or friends defines our mobility as well as the other way round, as we may make new friends by visiting new places. Opportunistic networks directly support this local social interaction [78]. Opportunistic network based local social networks do not have privacy relevant information stored on a central server. Furthermore, presence awareness of friends nearby [98] and finding new people with similar interests [1] is easily supported.

Be it enhancing existing services or providing alternatives and even new services, the scenarios and applications described in this section show the potential for opportunistic networking. While the potential is recognized, implementing opportunistic networks in reality has experienced many challenges. One major challenge was the lack of suitable architectures for opportunistic networks. The following Section describes opportunistic networking architectures that have been proposed in research, as well as some examples of commercial applications that use opportunistic networking techniques.

1.2 Opportunistic network architectures

The challenged nature of opportunistic networks, such as typically unconnected paths, prevent the Internet architecture from being applied. Consequently, a significant branch of research on opportunistic networks is dedicated to designing suitable architectures. One of the early occasions, where it became evident that the design decisions taken for the Internet are not suited for all networking situations was the idea of the interplanetary Internet [15].

In a scenario where a command center on earth communicates with a space craft orbiting a planet and a probe on its surface, the connection between the command center and the space craft is lost each time the probe orbits around the far side of the planet and the space craft only has contact with the probe on the surface when it passes over it. An architecture where disconnections are an error condition, like the Internet, is not suitable for such a scenario. Therefore, an architecture that expects random and frequent partitioning of the network, node disconnections and long data propagation delays is required. To this end, opportunistic networks and related fields such as delay tolerant networks (DTN) [23] and pocket switched networks (PSN) [16] have been investigated for more than ten years. Solutions for challenges such as sporadic connectivity, unknown neighbors and contact duration have been developed. While architectures are not at the core of this work, we survey some important opportunistic architectures that were proposed. To locate our contributions in the context of opportunistic networking, we will then identify common features among all architectures, which we will call fundamental building blocks.

1.2.1 DTN

In 2003, the first architecture designed for challenged Internets, called DTN (delay tolerant networking) [30] was proposed. Since then, this effort was continued by the IRTF DTN research group [23]. The research group has published a series of RFC documents that further specify this architecture. Its main element is a *bundle* protocol, which organizes and exchanges messages in bundles that are specifically designed to cope with long delays and disconnections without data loss. The DTN architecture is designed for generic networking situations with long delays and disconnections. It thus does not take advantage of context specific knowledge for its proposed routing solution Prophet [64], such as the human mobility of nodes in opportunistic networks. Furthermore, it defines generic categories of contacts mainly based on their predictability, but deferring the detection of a contact to technologies below the DTN overlay.

1.2.2 Haggle

The Haggle architecture [81] organizes messages in application data units (ADU) which, similar to bundles in DTN, contain all required information to

be forwarded independently of other ADUs. More specialized than DTN, it is specifically designed for mobile device based opportunistic networking and to additionally utilize infrastructure networks. It supports different routing solutions, among one of them, BubbleRap [52], uses social information in its routing decision. Haggle also explicitly requires neighbor discovery, which includes finding opportunistic neighbors (e.g. by Bluetooth scanning) as well as fixed Infrastructure (e.g. by scanning for WiFi APs).

1.2.3 PodNet

PodNet [59] is an opportunistic architecture designed for content sharing based on a publish-subscribe paradigm. Content is organized in channels that users can subscribe to or create new channels themselves. Data forwarding decisions are based on the subscriptions and are controlled by the users seeking content (i.e. content is not pushed to other nodes without request). Neighbor discovery is also an integral part of the PodNet architecture and can rely on available technologies (e.g. Bluetooth and WiFi scanning) as well as an integrated beacon mechanism that is used in case the wireless technology used does not provide a neighbor detection mechanism. An extension of the PodNet architecture introduces additional security elements which limit access to the content channels [92] to publishing by authorized users only or to completely closed groups of users and a reputation system for content creators.

1.2.4 Scampi

The SCAMPI architecture [79] is designed as a service platform for opportunistic networks. It contains elements of the three aforementioned architectures and combines them to make flexible services (e.g. content or resources) available to opportunistic nodes. It routes self-contained messages similar to DTN bundles with a flexible selection of routing mechanisms along opportunistic contacts detected by the neighbor discovery.

1.2.5 Commercial applications

While the mentioned architectures did not successfully spread into practical use as general purpose opportunistic networking enablers, there are some applications available that use opportunistic networking based or enhanced

services for smartphones. One of the best known is FireChat [68], a messaging application by Open Garden that was used by the protesters in Hong Kong in 2014 [10]. Twimight [27], a twitter client that is able to propagate tweets opportunistically if the infrastructure network fails, is available for Android phones. An opportunistic VoIP application has been developed in the Serval project [91] to enable telephony in areas lacking cellular coverage. Uepaa [97] created an alpine safety application that maintains communication among users even if they venture beyond the reaches of cellular connectivity, which is available for iOS and Android. Tracking way points and rescue alerts are propagated opportunistically until they reach an area with cellular connectivity to aid in and speed up the rescue operation. More recently, Uepaa also provides its opportunistic technology as platform service for use in other applications [98]. Leaving the software only domain, goTenna [39] has developed an external device with a dedicated radio for opportunistic networking. The goTenna device works in combination with a smartphone to which it is paired using Bluetooth LE. The device contains a second radio interface that can detect and communicate with other goTenna devices up to a distance of several miles. This range is mainly due to its operating frequency at approximately 150 MHz [31], at which it benefits from different propagation properties than the typically used 2.4 GHz radios (Bluetooth, WiFi). As a consequence, goTennas opportunistic communication is exclusive to users with goTenna devices (the devices are sold in pairs only).

1.3 Fundamental building blocks in opportunistic network architectures

The architectures described in the previous section all have two significant features in common. The first, *neighbor detection*, establishes the basic unit of communication in the opportunistic network, the contact, by detecting other nodes within transmission range. In the second common feature, *routing*, a strategy to exploit the individual contacts to propagate messages from a source to a destination is defined. Thus, the two fundamental building blocks of opportunistic networks, neighbor detection and routing, provide the basic opportunistic networking features which enable the users to run the opportunistic application of their choice. Each of the contributions of this thesis is related to one of those fundamental building blocks. In the following, we give a brief overview about the fundamental building blocks before we will focus

on the use of social information in those blocks in the following Section 1.4.

1.3.1 Neighbor detection

Neighbor detection is the building block that provides all the functionality in order to detect other devices in vicinity and thus make contacts available for opportunistic communication. Neighbor detection mechanisms can broadly be separated into three categories. First, some wireless technologies (e.g. Bluetooth) come with an integrated neighbor detection that can be exploited. Second, specialized beaconing and scanning schemes are designed on top of wireless technologies without built in neighbor detection, or when the built in detection is for some reason not suitable for opportunistic networking (e.g. too slow). For example, WLAN Opp [93] uses 802.11 access point and station roles dynamically for each node to detect and establish connections to other devices in the vicinity. Also, PodNet uses its own beaconing scheme to detect other nodes connected to the same access point. A major challenge with neighbor detection mechanisms is the energy consumption or other resources, which may differ significantly across roles (e.g. WiFi AP and station [93]) or sent packets (e.g. broadcast and unicast). Third, mainly due to energy constraints on mobile devices, the use of additional, low power radios has been proposed [54]. This allows to optimize neighbor detection with minimal impact on the overall device run time. Either, existing radios in smartphones are used, such as a combination of Bluetooth (short range, low bandwidth, low power) and WiFi (long range, high power, high bandwidth), or a new low power radio that has similar range to the high power radio is used. Since this approach may require additional hardware, it is not within reach for opportunistic networks using off the shelf smartphones, until device manufacturers have an incentive to agree on a technology and implement it in their products.

1.3.2 Routing

Once a device is aware of other contacts, a strategy to transport information across devices is required to support data exchange. Traditional Internet routing does not fit the unpredictable and time varying nature of opportunistic networks. Even protocols that are designed for frequent disconnections (but where connectivity is the norm) such as AODV [75] or OLSR [19] do not perform well in opportunistic networks where disconnection is the predominant state. Thus, new routing algorithms are required. To route messages in

opportunistic networks, two strategies in its routing protocols can be identified. The first is to distribute *multiple copies* of a message and the second is to exploit *structure in the contacts*. Distributing multiple copies (flooding) has been proposed with epidemic routing [100], which creates a copy of each message for all nodes that are encountered. The amount of messages created by flooding puts a strain on the resources of the network (e.g. bandwidth, storage), therefore, various ways to limit the number of message copies while maintaining message delivery rates were proposed (e.g. Spray and Wait [86]). More relevant for this thesis is the strategy to exploit structure within the contacts. The structure in opportunistic contacts is caused by the social behavior of people carrying the smartphones. How the social information is used for opportunistic routing is described in Section 1.4.2.

1.4 Social information

In the context of this work, *social information* is information about the relationships, interactions and behavior of its users that is made available, is used or collected by the operation of an opportunistic network, implicitly or explicitly. Furthermore, we narrow the focus to the use of social information in the two fundamental building blocks of opportunistic networks. In this sense, social information consists of information about who is in a social relationship with a given user, such as family, friendships, co-workers or familiar strangers. The operation of an opportunistic network makes this information available, so that network operations such a routing can benefit from it. In general, social information used in neighbor detection or routing can be of two different origins:

- Self-reported by the users.
- Implicit information (e.g. contacts) gathered during the operation of the opportunistic network.

Self reported social information requires the user to mark social links and potentially even assigning them to categories (such as friend, family, loose contact, ...). Implicit information is derived from observation of networking events (contact frequencies, times, patterns, ...) which have their origin in the social behavior of the users. For example, a frequent encounter with another node during office hours on work days indicates a contact to a coworker, while contacts in the morning or evening may relate to family members.

Social information is successfully used in the fundamental building blocks of opportunistic networks to improve networking performance and security, which we will survey in sections 1.4.1 and 1.4.2. In general, the social information is readily available for all potential networking mechanisms to make use of it. Unfortunately, it therefore is also available to any other interested party who can use it in any way it wants.

1.4.1 Social information in neighbor detection

In traditional opportunistic networks, nodes participate using their real identity. As opportunistic networking absolutely requires neighbor detection for its operation, the nodes identity is permanently broadcast. In such an opportunistic network, every node is aware of the identities of the other users around. While social information is not required for the purpose of neighbor detection itself, the availability of nodes identities is the basis for several security and performance improvements in opportunistic networks. Examples for those improvements are Sybil defense [94], spam prevention [95] and social based routing (see Section 1.4.2). In [107] the authors build on explicitly available social links to improve location privacy in a location based service. The principle idea behind the mechanisms based on social information are explicitly set up social connections (sometimes also called friendships). Once a social link has been created by two users, they can easily recognize each other based on the identities discovered during neighbor detection. By analyzing aggregated contact information, social structure can be reconstructed [9]. Therefore, the neighbor detection process is a source of social information in the opportunistic network. If neighbor detection does not deliver identity information about other nodes (e.g. they act anonymously), the mechanisms based on social links fail. Section 1.6.1 we will discuss consequences and strategies to protect identities and still benefit from social links.

1.4.2 Social information in routing

Opportunistic routing can greatly benefit from exploiting structure within human mobility and social behavior to route messages. As human mobility is influenced by people's social behavior, also the contact events in an opportunistic networks inherit the properties of the social behavior. Thus, the social information is used to predict, which node is better suited to carry a message towards a destination.

Social information consists of information about who is in a social relationship with a given user, such as family, friendships, co-workers or familiar strangers. Furthermore, social information about individual users can be aggregated into a graph [48] that gives more information about the social structure of a group of users. This aggregated information is even more useful to make routing decisions, as a user's utility to carry a message can be evaluated beyond one single hop in the graph. Such an approach can for example identify users that are part of two different communities and thus are well suited to carry a message from one community to the other. Section 1.6.2 gives a more detailed description of how this information is used by state of the art routing algorithms for opportunistic networks.

1.5 Privacy of social information

The most prevalent and successful business model for Internet services is free-of-charge services for the users, financed by advertising. Advertisement revenues make up the largest fraction of Internet revenues and the more targeted specific ads can be placed, the higher their impact and thus the revenue. To deliver ads specifically to a target audience, information about Internet users is collected in order to classify them into groups. Gender, age and other information used for this classification (up to 50 criteria or more) is privacy relevant. Thus, private information becomes the currency that Internet users pay for "free" services [57]. Furthermore, the rising popularity of social networks has launched a debate about freely giving away privacy relevant information online. This debate has raised user awareness and a growing number of users actively manage privacy settings on websites like Facebook, Twitter or Flickr. Also, events around the publication of NSA documents by Edward Snowden have shed light on the possibilities of Internet surveillance and the question what information should be private in general has become an important part of the public discussion.

Users today evaluate new technologies more critically than ever regarding privacy. Since opportunistic networks are not widespread in use today, privacy concerns need to be addressed from the beginning when introducing this technology. [63] gives an overview over privacy challenges for opportunistic networks. In opportunistic networks, the relationship between a device and its user is very close, in some respects even almost identical. This increases the scope of privacy concerns beyond what is typically found in Internet applications, as information made available by the user and information created by

the operation of the opportunistic network can combine in a way that reveals much more about users than in the traditional Internet setting.

Privacy in general is a broad topic where networking is only one element. Already for opportunistic networks, there are many privacy aspects that come into play. Opinions and preferences of the users or personal data contained in content that is exchanged are just two examples. As opportunistic networks are driven by user mobility, the users location is another aspect of privacy that is also related to neighbor detection. Announcing the devices presence also reveals a users location, which creates an additional incentive to act anonymously and unpredictably.

1.6 Related research

This section discusses existing research related to social information and privacy of social information in the fundamental building blocks of opportunistic networks. In Section 1.6.1 we discuss existing work that is related to privacy of social information in the process of neighbor detection as well as computing techniques that are available to use social information in a privacy preserving way. Section 1.6.2 describes how social information is used in state of the art opportunistic routing and graph modification techniques that are used in this work to implement privacy protection for routing. The two sections following discuss and compare an existing privacy protected routing approach with this work.

1.6.1 Privacy protected social link detection

With the introduction of privacy protection in neighbor detection, nodes essentially act anonymously. While this protects the nodes identities, it has consequences on networking mechanisms that are based on social information provided by the neighbor detection (already described in Section 1.4.1). A basic solution idea is, that the two nodes that come into contact somehow determine if they share an existing social link (e.g. the have established a friendship earlier) without revealing information about their identity or social links if this is not the case. To solve this dilemma, two basic tool sets are available. The first, *privacy preserving matching*, is based on cryptographical computations that allow the nodes to determine if they share a common attribute (e.g. a social link) without revealing their attributes to each other. The second, *hash based compression and comparison*, uses hash algorithms

to obfuscate the nodes attributes (e.g. their social links) and determine with a certain likelihood if they share a common attribute.

Finding common information among two nodes in a privacy protected way is achieved by research in the domain of privacy preserving matching or privacy preserving set intersection. A client-server based mechanism that allows only the client to learn the result of a computation that involves input from both was proposed in [34]. However, this mechanism requires a role determination for the client and the server, which increases its cost for application within opportunistic neighbor detection. It would also need to be run in both directions, which further increases its time and resource consumption. Efficient private set intersection is proposed in FindU [102] to determine the best matching user given a group of users and their attributes, without revealing the attributes of each user. The authors also propose a variant called private cardinality set intersection, where only the number of common attributes of the best matching user is revealed instead of the attribute values. The main limitation for such an approach in neighbor detection is, that it requires more than two users for the computation which can not be satisfied if only two users in an opportunistic network meet. Finding new friends by calculating social proximity instead of detecting existing social links is proposed in [21]. The authors define social coordinates for each user and calculate the distance between two users to determine whether or not they should become new friends. Their scheme however, relies on a central trusted server that computes the social coordinates for each user in regular intervals. Privacy preserving matching in general has a high computational demand that makes it unsuitable for a situation such as neighbor detection, where social link detection has to be done frequently at the beginning of each data exchange and the contact time is a scarce resource.

The second tool is the Bloom filter [11], which is a hash based structure that serves as a compacted representation of a set of attributes. Bloom filters are used as a tool in various scenarios. We focus here on the use of Bloom filters used for privacy preservation or in the context of opportunistic networking. For E-SmallTalker [106], Yang et al. use iterative Bloom filters for common interest detection. They use the Bloom filter as a compression tool to fit a list of the user's interests into Bluetooth service discovery packets that are size limited. Cryptographically secure bloom filters have been proposed by Nojima and Kadobayashi [67]. Unfortunately, the computational complexity of their approach is too high for implementation on smartphones, which are the typical opportunistic networking devices.

1.6.2 Social information based routing

When users explicitly provide social information to the routing process, they make available who they are friends with. This information may be taken from existing online social network accounts. Self-reported social network routing (SRSN) [8] uses Facebook friendships to route information in an opportunistic network. People Rank [65] also supports self reported social links to assign a utility value to nodes for forwarding messages. With SSNR and OSNR [73] Parris and Henderson present opportunistic routing based on self reported social links. Specifically, the authors obfuscate the self reported friendships to protect users privacy. We discuss the relationship of their work with ours in more detail in Section 1.6.3.

The largest part of available state of the art routing protocols for opportunistic networks use information about contacts [20, 24, 52, 64]. Contact information is analyzed to extract the implicit social structure. Based on the extracted social information a utility value is calculated for each node. The utility value is then used to make the forwarding decision. Simple contact statistics such as the time since the last contact is used in the FRESH routing [24], where the utility of a node decreases as the time since the last contact increases. Thus, the node which was in contact with the destination more recently has the higher utility. While FRESH only looks at individual contacts, with PRoPHET [64] the implicit social structure of contacts was exploited for the first time. In PRoPHET, each node maintains a vector of utility values for all known destinations, called the delivery predictability. The delivery predictability is calculated based on the number of past encounters (frequency property), the duration since the last encounter (age property) and the values reported by other nodes for a given destination (transitivity property). This is already a first step to aggregate implicit social information for routing purposes, even though it exploits generic non-randomness in contacts and not human mobility and social structure in particular.

A direct analysis of the available social information in the contacts is made by Bubble Rap [52] and SimBet [20]. Those routing protocols aggregate contact information obtained from multiple nodes into a *contact graph* [48]. The contact graph consists of the nodes representing the users and the edges connecting the nodes encode the social structure. The performance of graph analysis based protocols such as SimBet or Bubble Rap depends to a large extent on how well the contact graph captures social information. How the contact graph can be created was investigated in [48, 50] and the authors showed that the contact graph can very accurately represent the social struc-

ture in an opportunistic network. The routing decisions are then based on utility values that are obtained by calculating graph metrics on the contact graph, such as degree centrality, betweenness centrality or similarity. Routing protocols such as PRoPHET, SimBet or Bubble Rap present the state of the art in routing for opportunistic networks. For example in Bubble Rap, each node is part of a densely connected community and the communities in turn are connected by bridge links, which is based on the idea that people are part of different social groups and the social mobility across groups is less than the connectivity within a group. Messages are first routed towards high betweenness centrality nodes, which allows the message to traverse the bridge links until it reaches the destination community. Once the message arrived in the destination community, it is forwarded to high degree centrality nodes, which are very well connected within the group and thus more likely to deliver the message to its destination.

Compared to routing based on user provided or device data, where social information is either provided freely by the user or not considered at all, contact graph based approaches reveal social information without user consent. Since contact graph based routing approaches have proven to provide the best performance, they are considered as state of the art by the opportunistic networking community. Consequently, we focus on protecting privacy of social information in contact graph based routing.

Relationship between routing and privacy protection

The process of contact graph based routing in opportunistic networks can be conceptually structured into three steps. First, collecting contact information on a single node. Second, exchanging this information with other nodes and building the contact graph. Third, making forwarding decisions using complex network analysis metrics of the contact graph (e.g. betweenness centrality). Our approach modifies the contact graph before it is exchanged with other nodes, after step 1. It is also applied after each exchange to allow for incrementally building the contact graph among nodes in a privacy protected way. This prevents other nodes from learning who is friends with whom (as would be visible in the real contact graph). To achieve this, we have to ask the question: *is it possible to modify (privacy protect) the contact graph without impact on the routing success?*

Graph modification

To answer the question raised in the previous section, the general research domain of graph modification supplies a set of tools that may be applicable to our problem. This section gives an overview over privacy protection and anonymization for social graphs, since it provides the background for our work presented in Chapter 2. For a more detailed discussion of social graphs and privacy related techniques, the reader is referred to Chapter 14 in [105] and Chapter 3 in [4].

The contact graph used by state of the art opportunistic routing is by design a social graph. The exchange of this graph with other nodes releases the graph, which therefore needs to be privacy protected. Generally, privacy breaches can affect three different elements of a graph: the nodes identities, their attributes and the links between the nodes. The contact graph is built to reflect the social links, therefore we will focus on the privacy protection of the links between the nodes. As the link structure is often also used to identify nodes, protecting the nodes identity is an implicit secondary goal. In our work, we do not consider node attributes and thus will not further discuss techniques related to attribute privacy.

The purpose of the privacy protected graph defines what properties of the graph should be preserved and which ones can be neglected. Properties to be preserved can either be of aggregated (e.g. sub graphs), spectral (e.g based on adjacency matrix eigenvalues) or topological (e.g. shortest paths) nature. The review of social information based routing in Section 1.6.2 shows that topological properties of the contact graph are exploited (as can be expected for the purpose of routing). Specifically, information such as node degree, betweenness centrality or similarity are often used. We therefore now focus on privacy protection for topological features of social graphs.

There are two basic approaches to preserve topological graph features for the purpose of this work. The first, *k-anonymity*, modifies the graph in such a way, that a given node can not be distinguished from at least k-1 other nodes in the graph, regarding a given feature. This represents a node centric approach to graph modification. The second, *edge randomization*, modifies the graph structure by randomly adding and deleting edges, which provides a probabilistic protection against re-identification. As our goal is to protect the social links (edges) in the graph, we use edge randomization as a starting point to protect the contact graph.

Random edge modification does not always satisfy the requirements of utility preservation in graphs. To this end, selective edge switches have been

investigated [45], where the sequence of switches is represented as a Markov chain. One switch operation takes two pairs of nodes that are connected and rewires the edges so that one node of each pair is part of one of the new links. After the switch, the original node pairs are no longer connected to each other. For our work, we relax the edge switching constraint to a free redistribution of a single edge in the graph.

1.6.3 Privacy protected social information based routing

With privacy-enhanced social network routing, Parris and Henderson first identify privacy issues related to using social information for opportunistic network routing. This subsection gives an overview about their approach and a detailed comparison with our work presented in Chapter 2 is given in Section 2.7. They define social network routing as "*one method to inform routing decisions. Making the underlying assumption that encounters between mobile devices are more likely to occur within groups of people who are connected to each other, for instance through friendship or co-location, than between random strangers, messages may be source-routed – forwarded selectively only between friends of the original sender.*" [73]. Parris and Henderson define their own routing mechanism that works according to their definition of social information based routing. The sender sends a message to one of his friends and attaches a list of his friends to the message. The message is then passed on as soon as one of the friends (from the attached list) is met, until it reaches its destination. Thus, a message can be transferred over multiple hops, but all hops have to be part of the message attached friend list. In their scenario, messages that are sent have a friend list (represented by hardware addresses) attached, which is used to route the message towards the destination. The authors identify this as a privacy issue, as the friend list is sent out in clear together with the message. They propose two solutions to prevent the friend list from being revealed: Statisticulated social network routing (SSNR) and obfuscated social network routing (OSNR). SSNR varies the size of the friend list that is sent out together with the message in order to hide the full real list from an attacker. It either adds non-existent friends or removes existing friends from the list. Thus, a local attacker can not learn the entire (correct) friend list from a single or few observed messages. OSNR aims at hiding the entire message-attached friend list from an attacker by inserting the hardware addresses of the friend devices into a Bloom filter with a salt. Forwarding nodes can query the Bloom filter for membership in the friend list of any node that they encounter. Both SSNR and OSNR can be combined.

The authors evaluate the routing performance by routing messages on three different traces, which are all available at Crawdad. The SASSY data set, which was collected using Bluetooth devices that were carried by 27 users at University of St Andrews in [8]. A sampled subset of the MIT reality mining data set [25] and a sampled subset of the data set collected at the University of Singapore [87]. The evaluation considers delivery cost, delay and ratio for the three data sets and the two approaches. They find that the OSNR obfuscation does not impact the SSNR results as long as the friend list is not increased to a size that exceeds the Bloom filter's maximum size for a given false positive rate. Varying the size of the friend list with SSNR does impact delivery ratio, where reduced list size increasingly reduces the message delivery rate. Messages are only forwarded according to the sender defined list, thus no further social properties (e.g. similarity) are considered by the forwarding nodes. The evaluation does not take messages into account, that are delivered directly from the sender to the receiver.

Parris and Henderson also discuss the security of their approach. Their attacker model assumes an attacker with limited resources that can only overhear communication sporadically. They state that "*the bar for an attacker has been raised significantly for reversing a single sender's friends list*" [73]. The authors evaluation of reconstructing the friend list gives an almost 100% probability of identifying a friend node after 9 intercepted messages. According to their analysis around 5 messages are required to brute force the Bloom filter as it is used in OSNR with Bluetooth hardware addresses.

1.7 Contributions and outline

In this work we investigate the use of social information in opportunistic networks from two different angles. First, we analyze how state of the art social based routing uses the available social information. Based on the analysis we show that the social based routing mechanisms collect and reveal more social information about its users than is required for the correct operation of routing. We therefore design and evaluate a mechanism to remove excess social privacy information in a way that does not impact the correct functioning of social based routing. Second, we show that increasing levels of privacy and anonymity negatively impact other features that are based on social links in the opportunistic network (e.g. security). To support interaction among friends that can improve security and user experience in an anonymous environment, we design a social link detection algorithm that works

in such an environment. The algorithm provides strong probabilistic protection of the existing social links and places only a small additional workload on the devices. Third, during the course of this work, we cooperated with a startup to implement a commercial opportunistic networking application on current generation smartphones. In this phase, we discovered that contacts we observed had a significant deviation from what was predicted by available research on mobility. We found that most studies considered the radio range to be a circle around a node. Effectively, only very little insight on how contacts are influenced by the smartphone design and the user body was available in literature. Both of those factors seemed to significantly impact the radio range. We therefore characterized all elements involved in the radio contacts from the smartphones components and design to the users' body in order to understand how contacts are influenced by those parameters.

In detail, we make the following contributions:

- We take a close look at state of the art social information-based routing using contact graphs. We find that the contact graph contains significant privacy relevant information that is not required for the correct operation of the routing. Thus, we design an algorithm that is able to remove surplus privacy relevant information from the graph and quantify the resulting impact on correct forwarding decision. The algorithm achieves this by selectively adding and removing edges in the graph. Even if the nodes only know a part of the graph, the privacy preservation provides good performance. We also introduce a variant of the algorithm that is able to effectively handle large graphs. Using privacy protected graphs to drive social routing, the message delivery is maintained very well. This part of the work was published in *Distl, B.; Hossmann, T., Privacy in opportunistic network contact graphs, IEEE, 15th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014*

- Privacy protection can have a negative impact, as social connections are hidden from the neighboring nodes themselves. We design a protocol that allows friends to recognize each other fast and efficiently in an anonymous environment. The protocol is based on a novel privacy preserving use of Bloom filters to prevent disclosure of a friendship "fingerprint". As contact opportunities are a scarce resource, the protocol is very lightweight and only consumes little contact time. We also implement and test the protocol successfully on the android platform. The research conducted in this part was published in *Distl, B.; Neuhaus, S.,*

Social power for privacy protected opportunistic networks, IEEE, 7th International Conference on Communication Systems and Networks (COMSNETS), 2015

- Contacts are an essential source of information. The impact of mobility on contacts is already well investigated. We take a new look at contacts from a point of view that was only little investigated so far. We characterize the effects of using smartphones as base opportunistic networking devices. That is why we ask the question "How well are smartphones suited for WiFi based opportunistic networking?". Instead of putting the focus on mobility and contacts, we focus on the properties of smartphones and the users that carry them. To answer the question, we look at all factors that influence the WiFi link budget in realistic scenarios. Our investigation gives clues about the minimum and maximum ranges for opportunistic contacts. This includes the effects of smartphone design and a simple body model. Based on available data, measurements and our model we also estimate the data transmission capacity of two pedestrians crossing in an outdoor area. This part of the work was published in *Distl, B.; Legendre, F., Are Smartphones suited for DTN networking? A Methodological teardown of smartphones' WiFi performance, International Workshop on Wireless Networks: Measurements and Experimentation (WINMEE), 2015*

1.7.1 Outline

The remainder of the thesis is structured into two parts. The first part contains two contributions for privacy preserving use of social information in opportunistic networks. The second part contains a contribution towards understanding how the radio properties of smartphones influence the contact events and thus opportunistic neighbor detection.

In Chapter 2 we present our algorithm to protect social privacy in contact graph based social routing. We first review the contact graph as state of the art routing information in Section 2.3. Next, we introduce the metrics we use to quantify privacy and the impact on routing accuracy. With random graph changes as a reference, we then present a step-wise optimal algorithm to change the graph with global as well as only restricted local knowledge about the graph in Section 2.4. As the contact graph size increases, we introduce an adapted algorithm that overcomes size constraints imposed on the step-wise optimal algorithm in Section 2.5. Eventually, we use the privacy

protected graphs to drive routing decisions on existing traces and evaluate the message delivery compared to using the original graph in Section 2.6.

How existing friends in an anonymous environment can be recognized is presented in Chapter 3. We review the privacy challenges in Section 3.2 and present our system and attacker model in Section 3.3. Afterwards, we design a protocol that detects existing friendships in an anonymous environment in Section 3.4 and give an estimation of the probabilistic privacy protection efficiency in Section 3.5 before looking at its performance on smartphones in Section 3.6.

In Chapter 4 in the second part of the thesis, we leave the topic of privacy and investigate the effects of smartphones and the people carrying them on the WiFi link characteristics. Motivated by significant differences between expected and measured contact distances we look at the smartphone link budget in Section 4.2 and the physical properties of current generation smartphones in Section 4.3. After revisiting the two-ray ground propagation model in Section 4.4 we present the results of our outdoor measurement in Section 4.5 and estimate the remaining elements of the link budget. Finally, we present a body model and its effects in Section 4.6 before investigating the range, data rates and possible data transfer volume of two pedestrians crossing outdoors in Section 4.7.

Part I

Privacy of social information in opportunistic networks

Chapter 2

Social link privacy for the contact graph

2.1 Introduction

The dynamic and random nature of opportunistic contacts poses many networking challenges. Problems like routing, resource allocation or service and content placement require rethinking the solutions designed for connected wireless networks. Also, different opportunistic networking applications are driven by different requirements. A first requirement may be to distribute information to all people within a geographic area, for example in the case of disaster communication. Flooding messages with the desired information for the opportunistic users can be an effective solution in that case. While flooding information is effective in such situations, other opportunistic networking scenarios can increase performance by exploiting further information for data distribution. In addition to optimize flooding, information based on the users of the opportunistic network can be employed to improve data distribution. Type of content and content popularity is an example that can be used in scenarios like cellular offloading. Above data distribution, routing in opportunistic networks can benefit from observing information available from the operation of the opportunistic network.

While social information is very useful, the implications of collecting and using this information on users privacy has not received much attention. The social information used, such as information about friendships and other so-

cial links, is often collected and distributed among the nodes. This leads to a situation, where privacy sensitive information about users is communicated throughout (parts of) the network. Each node then distills out the information required to make forwarding decisions. We investigate social graph based routing in opportunistic networks with regard to privacy sensitive information. We consider the individual friendships of users to be privacy relevant. The question that we will address in this chapter is: "How much information about social links can be removed without compromising social based routing?" This question can also be phrased differently: "How much accurate social information is required for social routing?"

The study of wireless contacts has become a popular field of research with interesting results [16, 56]. It was found that, since the contacts are driven by human mobility, they exhibit patterns akin to other social networks: contacts happen in *communities* [52] of people who see each other often (friends, colleagues, familiar strangers) with *bridges* between them [50]. To exploit these structural feature of contacts, it was proposed that nodes aggregate the history of observed contacts to a *contact graph*, which represents the structure of who is in contact with whom frequently, and use tools from complex network analysis [66] on this contact graph to make networking decisions exploiting its social structure. At the time of meeting, nodes exchange their views of the contact graph and hence get a more and more complete view of the structure of contacts.

For the example of routing, research results have shown that protocols that use *centrality metrics* computed on the contact graph show good performance. These protocols, which route messages greedily towards more and more central nodes towards the destination community show promising performance [20, 52]. In these approaches, when two nodes meet, they decide based on their centrality who has the higher utility, i.e., who is more likely to deliver the message or bring it closer to the destination. It is thus the *ranking* of the nodes with respect to their utility (e.g., centrality) that drives the routing decisions. Similarly, for the example of content placement, it was found that centrality can be used to efficiently distribute a limited number of content copies among the nodes [77].

Unlike routing in traditional connected networks, opportunistic networks do not allow for end to end routing decisions at a single point in time, due to the intermittent connectivity among nodes. Rather, upon node contact, an informed forwarding decision is taken by the involved nodes. The forwarding decision decides which node is better suited to carry on the data. The rout-

ing process is thus typically executed on each node and yields a forwarding decision for the current contact. While executing the routing process, nodes take various information into account, as required by the routing algorithm used. Social based routing on the contact graph uses the this graph as its main source of information. As links in the contact graph represent (stable) social connections, current contact might not be represented by a link in the graph. By evaluating the social structure regarding the current contact and the intended receiver, the graph can still be used to make a forwarding decision. There is thus no direct dependence between a contact used to forward data and a link in the contact graph. There is, however, a direct correspondence between the users of the network and the nodes in the contact graph.

While the contact graph is a promising tool to solve many challenges in opportunistic networking, it also poses serious privacy risks to the user. The contact graph encodes the history of who meets whom and how frequently, a good predictor of sensitive information like social ties, shared interests or personal communication [49]. Since the contact graph must be available locally at the nodes to decide which node has the higher utility in a meeting, this information is available to all nodes participating in the network, benign or malicious. We hence ask the question: *How can we prevent disclosure of links in the contact graph, while maintaining its utility?*

We therefore study graph change approaches that greedily modify edges in the contact graph, such that the ranking of nodes in terms of routing utility is maintained (thus not interfering with routing decisions). Using synthetic and measured contact graphs, we show that, assuming global knowledge, we can maintain the ranking of nodes for a large percentage of transformed links, thereby offering high levels of privacy. Yet, a characteristic property of opportunistic networks is that nodes only have access to local knowledge. In a second step, we propose and analyze heuristics by which the nodes can modify their local view of the contact graph and show that the global ranking is still largely maintained. To overcome scalability issues of this greedy approach in large contact graphs, we further propose a heuristic that selects the edges to modify, which is based on similarity values of two nodes.

Eventually we are interested in the impact of our privacy protection approach on real routing decisions. Thus, we route messages using real contact traces and existing routing protocols with the original and our privacy protected contact graphs and compare the results. Our evaluation shows that the routing performance even in the presence of our privacy protection scheme is good, while random changes have a significant negative impact on the routing

performance.

Summarizing, our contributions in this chapter are the following.

1. We show that the social information encoded in the contact graph can be protected with little impact on the ranking of the nodes regarding a routing metric.

2. A stepwise optimal greedy algorithm that performs very well, but does not scale to larger graphs.

3. A heuristic that approximates the greedy algorithm which performs well and is a scalable approach.

4. Even with only limited local knowledge, a distributed version of our algorithm still works well.

5. An analysis of routing performance indicates, that message delivery is very good, even if our privacy protected graphs are used to drive the routing decision.

The remainder of this chapter is organized as follows. In section 2.2 we survey related work in opportunistic networks privacy in general and social privacy in routing in particular. Section 2.3 introduces the contact graph as state of the art social routing information source as well as the metrics we use to capture social privacy and its impact on routing performance. Our algorithm is presented in section 2.4 and we evaluate the impact of our algorithm on our routing utility metric in section 2.5. Eventually we use the privacy protected graph to drive routing decisions with an existing social routing protocol in section 2.6 and present our conclusions in section 2.8.

2.2 Related work

Security and privacy issues in opportunistic networks have already been addressed from different angles. The authors in [14] exploit user mobility and thus contact patterns to set up security associations among users who are in contact. For this work, mobility (and the social structure within) is a key element, as it is also used to periodically renew the established security associations. To set up the security associations, communication over a secure side channel is used. In [83] the authors target content privacy directly. In the

context of content-based opportunistic networking, where source and destination are related to content instead of users directly, the authors define three privacy levels. The increasing privacy requirements of these levels are met by employing multi-layer commutative encryption.

In *secure discovery of neighbors* [71] the authors survey different neighborhood detection approaches and their security properties. While much of the survey is focused on physical effects and propagation, the authors point out the need for a more general approach towards neighbor detection and its security evaluation. Preserving privacy regarding the interests of users is proposed by the authors in [7]. They propose to use *private matching of shared interests* to detect new friends with common interests in a mobile social networking scenario. On another aspect, *privacy of context information* [70] the authors acknowledge and address user-centric context information. They propose to take the current user context into account when deciding which information to reveal and also to exploit contacts to privacy protect the information. Another approach to protect the identity of the sources and destination in an ad hoc network is proposed in [2]. Based on packet coding, which allows to combine the properties of multicasting and onion routing into one approach, the authors maintain sender and receiver anonymity. Obfuscating the relationship between sender and receiver is also done by [58], in which the authors use groups of users to reroute messages in combination with cryptographic tools to provide communication privacy in an opportunistic network setting.

Unlike these approaches, we look at the privacy of information used for routing. There are many examples where social information of some form is used in opportunistic networking, such as [8, 13, 64, 65]. Privacy of the information used for routing has not received much attention. Parris and Henderson [73] investigate the case of social network routing, where friend lists are exchanged among nodes to inform routing decisions. The authors obfuscate the friend lists and similarly to *Statisticulated Social Network Routing (SSNR)* proposed in [73], we aim at obfuscating the links in the contact graph. Parris and Henderson [73] analyze random addition and removal of links and its impact on routing performance for simple routing protocols.

Here, we take this a step further and investigate the impact when using complex routing metrics like betweenness centrality (often used to route between communities [20, 52]) in realistic graphs with community structure. For such metrics, we show that random addition and removal of links affects routing decisions already for far smaller percentages of changed links than

for simple routing protocols that are not based on social information. Thus, we need more sophisticated mechanisms for transforming the contact graph.

2.3 Algorithm elements

In this section we explain the three key elements in our approach: *opportunistic contact graphs*, *routing utility ranking* and *privacy*. Contact graphs are a powerful tool to capture a reliable link structure within an opportunistic network. Opportunistic routing compares utility values to decide whether data should be passed on to a node in range, so absolute values can be reduced to a ranking without impact on the routing. The contact graph contains information about social links that must be protected. Otherwise this social information is available in clear, compromising the privacy of opportunistic network users. To find a trade off between privacy and routing utility, all three elements (contact graph, utility ranking and privacy) need to be considered together.

2.3.1 Opportunistic network contact graphs

Opportunistic networks are formed by mobile devices with wireless capabilities (e.g. WLAN, Bluetooth). Whenever two of those devices come into communication range, they can exchange data directly without the requirement for any infrastructure (e.g. WLAN access points). The contacts are guided by the movement of the nodes and their radio range. Contact duration, sequence and frequency are stochastic and unpredictable. To make best use of the opportunistic network, a tool allows to identify a reliable structure within the stochastic contact events. The construction of contact graphs is well accepted as a tool and has proven to be useful in many challenges regarding opportunistic networks. The contact graph G consists of a set of nodes V that are connected by a set of edges E. Edges are stable links that were selected during the generation of the contact graph [50]. Those edges are used by routing algorithms to properly transport packets in the network.

In figure 2.1 an example of how an opportunistic contact graph could look like is shown. The edges indicate, that there seem to be stable links which can be used to route packets.

Contact graphs are an accepted and useful tool to perform various tasks in opportunistic networking. Using contact graphs, our approach relies on

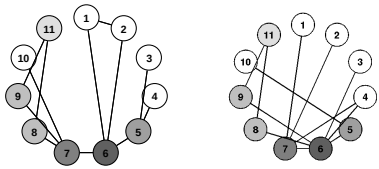


Figure 2.1: *Contact graph with (a) original structure and (b) with 30% edges changed by greedy privacy protection. Node colors reflect the betweenness centrality ranking of the nodes (darker is higher ranked).*

a proven tool and additionally, the remaining two elements of our approach, routing utility ranking and privacy, are directly related to the contact graph.

2.3.2 Routing utility ranking

One use for contact graphs are routing mechanisms in opportunistic networks. Often routing decisions are based on metrics of the contact graph. We argue that in most cases, only a ranking of the nodes regarding a metric is important for the routing decision instead of absolute metric values.

The nature of opportunistic networks make routing packets to a destination challenging. In classical routing, an end-to-end route can be determined by analyzing the connectivity graph of a network. In opportunistic networks, links are often only available for unpredictable and short periods of time Thus, it is impossible to calculate an end-to-end route. Routing decisions are made on each node encounter based on the information encoded in the contact graph edges (e.g social links). The metric used for this decision captures the probability of a node to be in the "correct direction" towards the destination.

Without additional information about the graph, a common approach is to forward packets to nodes with a higher node degree, with the assumption that those nodes have a higher chance of meeting the destination node. For example node 5 in figure 2.1 wants to send a packet to node 10. Node 5 meets either nodes 3, 4 or 6. It compares its degree with the respective node. In the case of meeting nodes 3 or 4 its own degree is higher so it does not forward the packet to any of them. When it meets node 6, which has a higher degree, the packet is forwarded. Node 6 keeps the packet until it meets node 7, which has an equal node degree and since node 6 did not meet node 10 itself, the packet

is passed on to node 7. Node 7 does not meet another node with higher node degree, so it keeps the packet until it meets node 10 and delivers the packet to its destination.

Several measures are used in proposed routing algorithms like centrality and betweenness. The next hop node is determined by comparing the metric (utility) values for each encountered node. Thus, the actual value of the metric is not taken into account by the routing algorithm, only the comparison results. In our example, node 5 has a node degree of 3. Nodes 3 and 4 both have a node degree of 1, so it does not forward them the packet, because $3 > 1$. Reducing the numerical values to a ranking does not influence routing utility but allows for more flexible changes in the graph, as only the $>$ relation among all nodes needs to be maintained.

Opportunistic routing mechanisms exploit the contact graph to achieve better performance by making use of edges that represent reliable connections based on social links. A ranking of the nodes regarding their utility in the graph is sufficient for the routing decision process. While contact graphs improve routing decisions, the encoded social link information also reveals privacy sensitive information about users, which is detailed in the following subsection.

2.3.3 Privacy of social links

So far we have considered mobile devices interacting with each other. In practice, most mobile devices are personal devices and a device and a user can be regarded as one combined entity. As a consequence, data sent by a specific device can be attributed to a specific user. Due to the graph generation process, edges in G now represent social links. In this subsection we will define our notion of privacy for the contact graph and detail an attacker model.

Social link privacy in our model is the ability to give users plausible deniability for the edges in the contact graph. That is randomizing their set of social links. Thus, in our attacker model, the attacker wants to guess, whether or not an edge between two given nodes in the visible (protected) graph exists in the original contact graph. For this the attacker knows:

- The privacy protected graph $G' = \{V', E'\}$.
- The privacy protection algorithm.

That is, given the privacy protected graph $G \rightarrow G'$ and a node pair $n1, n2 \in V$ with $e\langle n1, n2\rangle \in E'$ the attacker wants to find out whether $e\langle n1, n2\rangle \in E$.

Our attacker acts on a graph level and is not able to:

- Track the users movements.
- Use any additional background knowledge.

Given the attacker classification in [105] our attacker falls into the link disclosure problem category.

In the context of contact graph based routing, two main goals need to be addressed in a trade off. On the one hand, the social routing metric used to make decisions must be maintained in order to keep routing possible. On the other hand, as much of the encoded social information as possible should be protected. This is why plausible deniability for the edges in the graph is the goal of our algorithm. Attacks that aim at identifying the top ranking nodes regarding the routing metric will still succeed, as this is exactly the feature of the graph that our algorithm tries to maintain. The routing metric graph feature is at the core of contact graph based routing and therefore has to be available to the nodes. We can thus put the goal of our algorithm as a question: "How can the excess privacy relevant information be removed from the graph". Therefore, we strive at minimizing the privacy relevant information available to the attacker from the graph.

To measure link disclosure probability in a simple way, our privacy metric expresses how many of the originally existing links in the graph still exist after applying the privacy protection mechanism. Without additional information, this metric expresses the probability for the attacker to guess correctly whether two nodes have a link in the real graph or not. The goal of our algorithm is to preserve the routing utility ranking for the graph for a given metric. As metric values are only used in comparisons, our algorithm tries to maintain the correct ranking of the nodes instead of their absolute values. In that case the comparison gives the same results as before. Based on the classification in [105] our algorithm falls into the category that tries to achieve feature preservation in the graph with edge randomization, where our algorithm tries to find a trade off between randomization and successful feature preservation. From the threat analysis done in [73], our attacker wants to:

- Discover structural information about the contact graph.
  - learn if a friendship link exists between two users.
  - learn how many friendship links a particular user has.

Our privacy protection approach does not attempt to be ready for implementation in a real system. The goal is to understand the characteristics and trade off involved when changing the contact graph to increase privacy, while at the same time keep the graph usable for opportunistic routing.

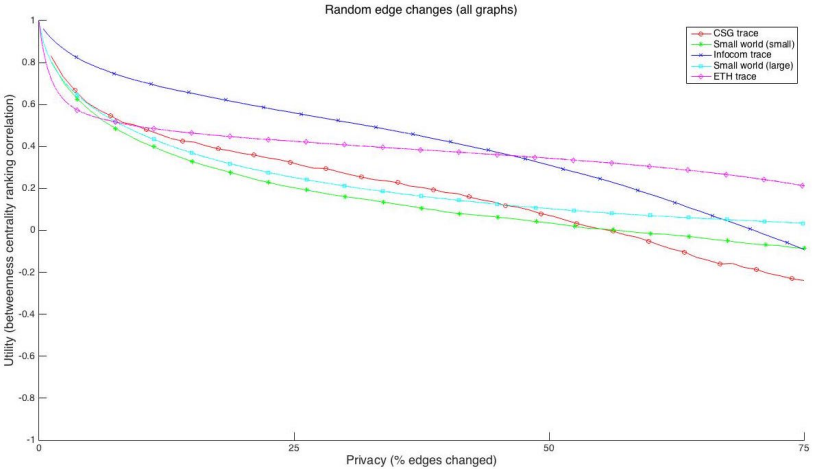


Figure 2.2: *Random edge changes for all graphs*

After introducing all metrics and the idea of the algorithm, we can first look at what happens to the metrics if changes in the contact graph are done randomly. Figure 2.2 shows the result for random edge changes for all graphs described and used later in the evaluation in section 2.5. The curves shown are the mean of multiple runs (100-1000 depending on the graph size). The x axis shows the increasing privacy protection towards the right. The y axis shows the corresponding routing utility rank correlation for a given privacy level. Note that at a correlation of 0, the the routing decision of two nodes will only be as good as guessing. Goal of our algorithm is to provide higher than random utility rank correlation for all privacy protection levels.

In this section we have shown how opportunistic contact graphs capture the social links of users to provide a basis for routing decisions. On this graph nodes can be ranked according to their utility for routing packets in the opportunistic network. While correct routing is required for opportunistic networks, without intervention the social links of each user are directly visible to anyone. This privacy issue can be addressed by changing links in the graph

in a selective way that maintains the routing utility ranking of the nodes but makes it difficult for an attacker to discover real social links between the nodes.

2.4 Approach

This section describes our approach to changing the contact graphs in a controlled way. Our basic idea is to add and remove edges in a sequence that maintains the routing utility ranking in the graph. Different edge selection strategies, greedy and heuristic, that can be used to determine this sequence are introduced. They differentiate in effectiveness as well as computational complexity. Both strategies can operate on either the entire graph or only a local subgraph (i.e. the direct friends of a node), which reflects the fact that individual nodes usually do not know the entire graph.

2.4.1 Metrics

There are two metrics that are used during the evaluation of the algorithm variants. One metric, utility rank correlation, is used to measure the impact on the routing decisions of the graph. Another metric, edge change percentage, quantifies the privacy of the graph.

Routing utility metric

One aspect of the algorithm is to maintain routing utility ranking in the graph. The algorithm itself is defined independently of any specific routing metric. For the evaluation, betweenness centrality is selected as the base routing metric on the contact graphs [35]. If every node is connected to every other node via the shortest possible path, betweenness centrality for a node indicates how many of those shortest paths go through this node. The more shortest paths a given node is part of, the higher its betweenness centrality will be. Betweenness centrality is used as metric for forwarding in [20, 52] and is defined for node v as in equation 2.1.

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{2.1}$$

Where s and t are two nodes in a graph, σ_{st} is the total number of shortest paths between the nodes and $\sigma_{st}(v)$ the number of shortest paths that pass through node v.

As described in the introduction, the usual routing decision on a contact graph is done when two nodes meet (come in contact) and have to decide which one of them is better suited to transport a packet towards a destination. The two nodes will compare their (possibly estimated) utility, which in the case of the evaluation is their betweenness centrality value. The result of the comparison is the decision which node will continue to transport the data. To maintain the correctness of this decision, only an ordinal ranking regarding the nodes betweenness centrality values is required.

Since our privacy protection algorithm (described in Section 1) operates stepwise, an evaluation needs to quantify the change in the routing performance for each step. As only the ranking of the nodes regarding the base metric are of interest, the original ranking and the ranking after each change need to be compared. The effect on the ranking is measured by the correlation of the utility rankings before and after an edge add-and-remove step. Since we only consider rankings, Kendall correlation, which is a rank correlation, is used. It is defined as in equation 2.2.

$$\tau_B = \frac{n_c - n_d}{\sqrt{(n_0 - n_1)(n_0 - n_2)}} \tag{2.2}$$

Where n_c is the number of concordant and n_d the number of discordant pairs. n_0 is the number of all possible edges and n_1 and n_2 are counts for tied (equal) values in the rankings. The values for the utility rank correlation are between -1 and 1 where 1 means perfect match of the rankings and -1 a completely reversed ranking. A correlation of 0 means that the two rankings are independent from each other. The best result is a ranking correlation that stays at 1 or as close to it as possible. For our purpose of maintaining the routing performance, we are interested in the ratio of correct to incorrect routing decisions. Intuitively that means that at a correlation of 1 the routing decisions even in the presence of the privacy protection mechanism will always be correct. When the correlation decreases, there will be more and more mistakes in the decisions. At a correlation value of 0 the decision process is as good as guessing (for two nodes that want to compare their utility values) and at -1 the decision will always be wrong.

Note that the evaluation of the utility ranking takes place for every edge selection strategy used within the algorithm. Only the greedy algorithm (Al-

gorithm 3 in Section 2.2) additionally uses the utility metric in the edge selection process itself. That means that, while random and heuristic changes do not require specific knowledge about the utility metric, the greedy edge selection needs to take this information into account.

Privacy metric

The attacker's goal is to find out, whether two nodes have a friendship connection, represented by a link in the contact graph. Privacy in this context can be defined as the difficulty for the attacker to determine whether the link in the graph really exists or not. Another way to look at this is the probability for an attacker to guess correctly whether a link that she sees in the graph actually reflects a friendship connection. For the unchanged graph, this probability is always 1, as the attacker always sees the correct contact graph. The algorithm now changes edge after edge, making it increasingly difficult for the attacker to determine whether a link in the current graph has a corresponding link in the original graph. A simple measure to quantify this uncertainty for the attacker is the *edge change percentage* as compared to the original graph.

Note that as defined in section 2.4.2 the algorithm maintains the edge density, so the attacker always sees the same amount of edges in the graph. Furthermore, contact graphs generally are rather sparse graphs. That means that the attacker basically can ask two different questions:

- Does an observed edge also exist in the original graph?
- Does an observed nonexistent edge also not exist in the real graph?

The privacy metric we use directly corresponds to the first question and quantifies the probability for the attacker to guess correctly whether an observed edge also exists in the unchanged graph. We evaluate edge change percentages from 0 up to 75% of the original edges. For each edge that is removed, a non-existing edge insertion is required. This leads to a restriction of the edge density that the original graph can have, which is dependent on the target edge change percentage ($tecp$). The maximum edge density d for a given target edge change percentage $tecp$ is: $d \leq \frac{1}{1+tecp}$

2.4.2 Basic algorithm

The algorithm takes a contact graph G which is unweighted and undirected as input, and outputs a modified contact graph G' with a predefined edge change

percentage. The algorithm transforms the set of Edges $G = \{V, E\} \rightarrow G' = \{V, E'\}$ with $|E| = |E'|$.

The algorithm always combines two operations in one step:

- remove an existing edge from the real graph;
- introduce a new edge between two nodes which have not yet been linked by an edge.

That also means our algorithm will never add back an edge that existed originally in the graph. This makes guessing an existing social link in the real graph harder for the attacker with each step. Given the attackers goal to identify real edges in the obtained graph, allowing edge reuse would fail to reduce the attackers change to guess correctly. In the worst case, the algorithm would keep removing and re-adding real edges, which would result in no privacy protection at all. Pseudo code is given in algorithm 1.

The algorithm does not indicate any "ideal" edge change percentage, but leaves the choice of the target percentage to the application. This trade off decision will depend on the application requirements for routing utility vs. privacy. The evaluation of our algorithm in section 2.5 will provide insight in how to set the edge change percentage to achieve the desired characteristic.

Algorithm 1: Basic algorithm

Data: E = set of existing edges in real graph
Data: $\overline{E}$ = set of non-existing edges in real graph
Data: E' = set of existing edges in changed graph
Data: $\overline{E'}$ = set of non-existing edges in changed graph
Data: ecp = percentage of real edges changed
Data: $tecp$ = target edge change percentage

0.1 **while** $ecp < tecp$ **do**
0.2 e,$\overline{e}$ = `selectEdgePair`;
0.3 $E' = E' - e$;
0.4 $\overline{E'} = \overline{E'} - \overline{e}$;

2.4.3 Edge selection strategies

How the edges e and $\overline{e}$ that will be added and removed in each step are selected is the crucial decision for the success or failure of the algorithm itself. This subsection describes three edge selection strategies with different properties. Random selection serves as a reference for comparison, greedy selection tries to achieve the best possible result and heuristic selection aims at larger graphs, where more intricate strategies fail due to their computational cost.

Random edge selection

Random edge selection serves as a reference point to compare the other edge selection strategies with. In each step, the random edge selection chooses one edge e to remove that exists in the original edge set E and has not been removed in a previous step (see Section 2.4.2). The edge $\overline{e}$ to add is chosen from all non-existing edges in the original set $\overline{E}$ minus the edges that have been added in the previous steps. Algorithm 2 shows simplified pseudo code for the random edge change selection algorithm.

Algorithm 2: Random edge selection

1.1 select random edge e from $E \cap E'$;
1.2 select random edge $\overline{e}$ from $\overline{E} \cap \overline{E'}$;

Stepwise optimal greedy edge selection

In contrast to random edge changes, greedy edge selection tries to stepwise optimize the combination of edges to add and remove. In order to achieve this goal, the greedy selection strategy needs an additional piece of information. It needs to know which metric is used in the routing process.

Knowing the metric used, the stepwise optimal greedy selection strategy can evaluate all possible options for the next step. From this evaluation it selects the edge add and remove combination that has the smallest impact on the ranking of nodes with regard to the routing metric. To assess the impact, the stepwise optimal greedy selection evaluates the ranking of all possible edge combinations against the ranking of the original graph and picks the best match.

Algorithm 3 shows the pseudo code for greedy edge change. Since the algorithm generally does not depend on a specific metric, the generic function `utilityMeasure` is a place holder for the effective metric function used.

Algorithm 3: Greedy edge selection

```
2.1 for edge e in E ∩ E' do
2.2     for edge ē in Ē ∩ Ē' do
2.3         E' = E' − e;
2.4         Ē' = Ē' − ē;
2.5         utilityMeasure(G, G');
2.6         E' = E' + e;
2.7         Ē' = Ē' + ē;
2.8 e,ē = max(all utilityMeasure);
```

Depending on the utility measure used, the greedy algorithm shown here does not scale well to larger networks for two reasons: First, for each existing edge all possible edges that can be inserted must be evaluated. The final decision is done based on the best measure of the forwarding utility with the real graph from all edge combinations. That means the utility measure for each edge remove/insert combination has to be calculated which increases drastically with more nodes in the graph. Second, with betweenness centrality as routing metric for the evaluation, shortest paths calculation is involved. While it is known which shortest paths will be affected by removing a given edge, it is impossible to decide which paths will be shortened by inserting a new edge in a graph. As a consequence it is not possible for this measure to precalculate or differentially calculate the changes to the graph.

Heuristic edge selection for betweenness centrality routing

Depending on the utility measure used and the size of the graph, it is not always feasible to calculate the prospective change in utility for all edge combinations. A heuristic that captures some of the information and provides a faster way of deciding on an edge add-and-remove combination is required especially for larger graphs. This metric will need to make some assumptions about the nature of the utility measure used and the structure of the graph it is applied to.

The routing metric used for evaluation in section 2.5 is betweenness centrality, hence the heuristic developed in this section tries to heuristically approximate the edge selection as it would be done based by the greedy selection with betweenness centrality.

Often contact graphs have a community structure where groups of well connected nodes are connected with fewer bridge links to each other. In our definition of social privacy, an attacker wants to guess social links / friendships. As communities are by definition well connected (but not fully connected), changing links within a group can to some extent obscure the social links without influencing overall forwarding performance in the graph extensively. One possible heuristic that can be calculated easily and allows to model this concept of social links in well connected groups of nodes is node similarity.

Node similarity is a measure that indicates how many neighbors two nodes have in common. Similarity for a node pair thus is given by their common neighbors and the total aggregated number of neighbors those two nodes have. Two nodes A, B with neighbor sets N_a, N_b have a similarity defined by equation 2.3.

$$S = \frac{|N_a \cap N_b|}{|N_a \cup N_b|} \tag{2.3}$$

If a node pair does not have any common neighbors their similarity is defined to be 0. With this definition, the similarity is normalized to a value between 0 and 1 and can be easily compared. Based on the previous notion of well connected groups, it can be expected, that similar nodes will often be found within the same well connected group. Using similarity as the heuristic we can now define a new algorithm that uses this heuristic as metric.

The algorithm that makes use of the similarity heuristic first calculates the similarity for all node pairs in the graph. As the similarity only requires knowledge of the direct neighbors of each node, this can be done for large graphs. Then, the algorithm will select a node pair which has the highest similarity. In the real graph this node pair has to be connected and this connection must not yet have been removed by an earlier step in the algorithm. Next, a node pair which has the highest similarity and which is unconnected, is chosen. Unconnected in this context means that the node pair is not connected in the real graph and that no link between the two nodes has been inserted by a previous step in the algorithm.

Algorithm 4: Heuristic edge selection

Data: L = all node pairs
Data: sC = similarities of connected nodes
Data: sD = similarities of disconnected nodes

3.1 **for** *all* $l \in L$ **do**
3.2 **if** $l \in E \cap E'$ **then**
3.3 $sC(l)$ = `getSimilarity(`l`);`
3.4 **if** $l \in \overline{E} \cap \overline{E'}$ **then**
3.5 $sD(l)$ = `getSimilarity(`l`);`
3.6 e = `max(`sC`);`
3.7 $\overline{e}$ = `max(`sD`);`
3.8 return e,$\overline{e}$;

2.4.4 Global and local knowledge

The algorithm defined so far is able to operate on any graph. Whatever graph is fed to the algorithm might depend on the requirements but also on the available knowledge of the graph. In some situations the entire graph, in others only a part of the actual graph might be available.

Global knowledge

With global knowledge the entire graph with all nodes and links can be used by the algorithm to determine the next best step. This is the maximum information the algorithm can have.

Local knowledge

With local graph knowledge, the available information approximates what nodes in real opportunistic networks know about the contact graph. This knowledge consists of a local subgraph of the entire contact graph centered around a given node. The subgraph contains only the first and second degree neighbors of the center node together with all the edges between those nodes.

2.4.5 Approach summary

The approach to change the contact graph in a controlled manner is mainly influenced by two aspects. First, the strategy to select edges to add and remove depends on the computational complexity of the selection strategy. The computational complexity will set a limit to the size of the contact graph that the strategy can be applied to. In general, more demanding (e.g. greedy) selection strategies will only be applicable to smaller graphs, though are expected to yield better results. The larger the contact graphs become, the simpler the selection strategy needs to be. This allows for edge selection in large graphs, but is expected to come at the expense of effectiveness regarding the routing utility ranking of the graph compared to a greedy approach.

Second, the contact graph that the algorithm operates on is either the global (full knowledge) graph or only a local subset of the nodes and edges of the entire graph. Global graph knowledge is expected to provide better decisions for each algorithm step, as the absolute impact of each step can be measured accurately. Local knowledge more reflects the situation of real nodes in the network. They will know only a subset of the contact graph centered around them. With this limited knowledge, it becomes more challenging to make "correct" decisions for each step, as only the local effects can be evaluated for each step.

2.5 Evaluation

In this section we present the evaluation of our algorithm to improve contact graph privacy. First, we detail the graphs that we used during our evaluation, which include trace- and model-based graphs, and state our expectations towards the algorithm. Next we run all algorithm variants on each graph and present one plot per graph that includes all algorithm variants. We then discuss the plots for each edge selection strategy, to observe if the strategy yields consistent results with the different graphs. Not all edge selection strategies are feasible (computable) for all chosen graphs. Wherever feasible, all variants of edge selection are evaluated with global as well as local knowledge.

2.5.1 Model and trace graphs

The algorithm in general accepts any undirected graph, but is tailored towards giving a good solution for graphs that have properties that are expected from

contact graphs as described in 2.3.1. This section gives an overview of the models and traces used to generate the contact graphs for the evaluation. All graphs used were unweighted and undirected.

Small-world model

Contacts in opportunistic networks happen as users roam around, which they do according to their routines and habits. Thus, they tend to meet some users more frequently than others, which introduces structure in the graph. To model this structure we use a graph model that reflects the structure in contact graphs. The small-world graph model connects each node to a given number of its direct neighbors. From this first structure, a certain percentage of links are rewired to random nodes in the graph. Those rewired links effectively provide shortcuts in the graph that model shortcuts created by social links of users (e.g. when a user has two groups of friends). Such shortcuts can be used to improve forwarding and are thus an important structural factor in the contact graph. Small and large graphs were configured to approximate the corresponding trace graph edge densities. Small graphs were constructed with 20 nodes, where 4 neighbors are connected and a rewire probability of 0.2. Large graphs were constructed with 482 nodes, 19 neighbors connected and a rewire probability of 0.15. Per size, 500 graphs were created with the model.

Traces

The second source of contact graphs we can use are contact graphs that are generated by traces. Traces are a collection of contact events, indicating the involved nodes and the time and duration of the contact. Such a trace can be transformed into a contact graph as described in [50]. In contrast to model generated graphs, traces are always collected within a predefined setting, using a specific communication or data collection technology and are also prone to errors in the collection process. Nevertheless, they are a valuable source of information as traces try to capture what is really happening including all practical issues that arise during the collection. Another drawback is, that traces only allow to generate one (or very few) contact graphs and thus a statistical evaluation is limited. For our evaluation, only the local graph knowledge case allows for multiple runs on the same graph, as different sequences of center nodes for the local evaluation can be evaluated. In the global case, only ties in the ranking or similarity can be evaluated in different sequences.

As the observed variation in that case is very small, it is not shown in the plots.

Two traces are used in the evaluation to generate contact graphs. A smaller trace collected at Infocom 2005 recording Bluetooth contacts of 41 participants of the Infocom conference in 2005. This trace contains a total of more than $22'000$ contacts [16]. A second trace is derived from WLAN access point associations at the ETH campus [96] during almost 15 weeks between Oct. 25th 2006 and Feb. 3rd 2006. We choose the 482 users which associate at least 5 out of 7 days on average and pre-process the trace for short disconnections as well as the ping-pong effect, where devices jump back and forth between different APs in reach. We assume that two nodes are in contact when they are associated to the same AP at the same time.

2.5.2 Evaluation results

Changing the contact graph will have an impact on the routing utility. The goal of the evaluation is to assess, whether the greedy and heuristic edge selection variants can outperform the random edge selection. It is expected that a greedy edge selection algorithm with global knowledge will provide for a significantly better result than random. The impact of reducing the knowledge from global to local is expected to result in a stronger impact on the node ranking and thus on the evaluation metric. The design goal of the heuristic is to approximate the greedy outcome with a significantly reduced computational burden. Therefore, the heuristic is not expected to produce results that are as good as the greedy selection. Simulation was done in Matlab and all PRNG settings were documented.

General plot structure

Figures 2.3, 2.4, 2.5, 2.6 show the results for all algorithm and knowledge variants per graph source. The plots of all graphs have the same structure. The x-axis shows the privacy protection level that increases up to 75 percent of graph edges changed. In general the range for the privacy protection can range from 0 (no protection) to 100 percent (all edges changed). However, we limit our evaluation to a maximum edge change percentage of 75 percent. Selecting a desired privacy protection level is up to the designer of a system that uses the algorithm. The y-axis shows the utility value (utility rank correlation, see Section 2.4.2) that indicates how well the modified graph still reflects

the original ranking regarding the routing metric. Values for the utility rank correlation range between +1 and -1, but for visibility, we only plot the range from 0 to +1, as we want the modified graphs to have a utility that is as high as possible.

All graphs show the results for random edge changes as well as the heuristic with local and global knowledge (see Section 2.4.4). Due to the computational complexity, we can show the results for the greedy edge selection only for the smaller graphs.

All curves have different colors and marker symbols as indicated by the legend in the plots. Furthermore, the algorithm variants that are not deterministic (random and local knowledge) have error bars at ± 1 standard deviation. The sole reason for different locations of the error bars of different curves is to prevent overlapping for better visibility. Note that algorithm variants with global knowledge are deterministic and therefore do not show error bars.

Greedy edge selection

First, we concentrate on the greedy edge selection variant. As the greedy algorithm uses betweenness centrality as utility metric, it can only be applied to small graphs. We will therefore look at Figures 2.3 and 2.4 which show the results for the small-world model based graph and the Infocom trace based graph respectively. In both graphs, the greedy algorithm with global knowledge has excellent performance with a utility correlation close to 1 over the entire range of edge change percentages. That means, that routing decisions will almost always be correct in that case. The greedy selection strategy with only local knowledge also provides good results with a utility correlation between 0.9 and 1 over most of the privacy protection range.

Due to the properties of the stepwise greedy algorithm, which only looks at one step after the other, it sometimes makes ”mistakes” from a wider point of view. Since each step consists of one edge removal and one edge addition, it is possible that some steps end up with a higher correlation value than before, if the structure with the newly introduced link creates a better utility ranking than before. This is expected behavior of a stepwise greedy algorithm. The variations can be seen in the small up and down changes of the correlation values, which are superimposed on the general trend.

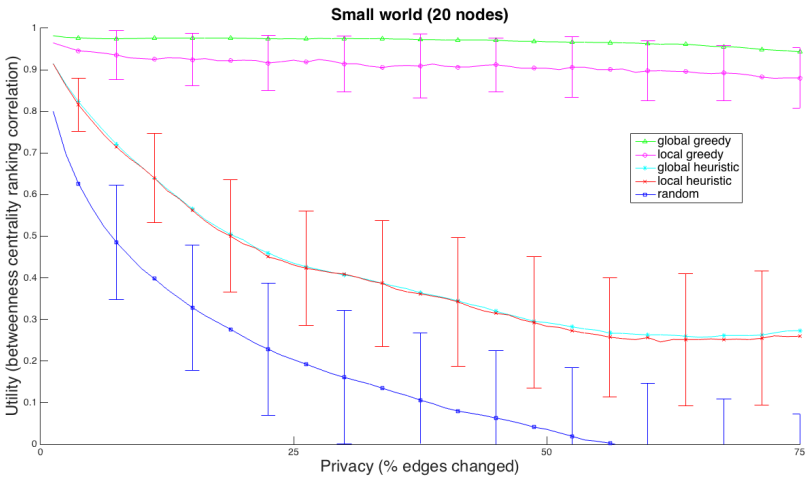


Figure 2.3: *Small-world model (20 nodes)*

Heuristic edge selection

With increasing graph size, the heuristic edge selection remains the only viable option, as the greedy selection can not be calculated any longer. The plots for all graphs (Figures 2.3, 2.4, 2.5 and 2.6) include results for the heuristic edge selection. However, we first focus on the results for the larger graphs of the small-world model and the ETH trace, as this is what the heuristic was designed for. The results for the small-world graph (Figure 2.5) and the ETH trace (Figure 2.6) show, that the utility rank correlation stays significantly above random changes for both graphs even for large edge change percentages. At the point where half of all original edges have been changed, the heuristic keeps the utility at approximately 0.75 for both graphs, whereas the random changes lower the utility to approximately 0.35 for the ETH trace and 0.1 for the small-world model. Note how close together the performance of the local and global knowledge heuristics are for the small-world model. This indicates that there is no difference in the ranking correlation errors introduced by local and global heuristic for large small-world graphs. In the case of the ETH trace, the local knowledge variant does not achieve a performance that is so close to the global knowledge, but still significantly above random graph changes.

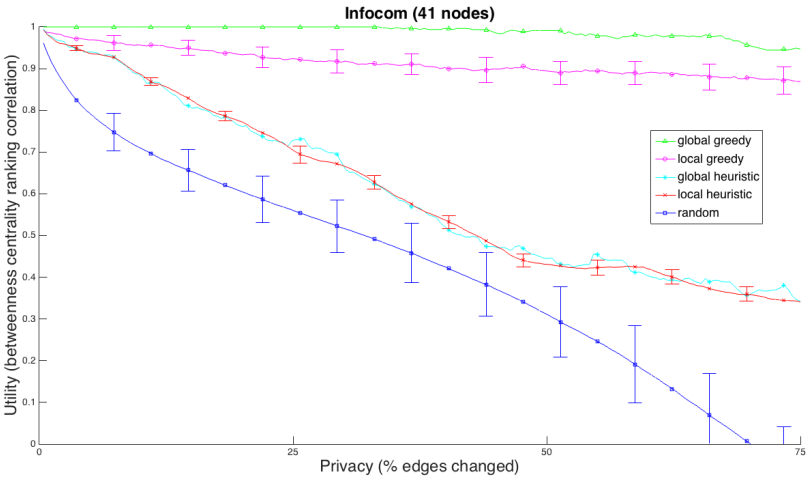


Figure 2.4: *Infocom trace (41 nodes)*

The almost equal performance of local and global knowledge for small-world model based graphs is caused by the properties of the graph model. The model creates explicit local structures first before rewiring some of the intra community links to bridges between communities. Since only few links from each community are rewired, the most similar nodes that are picked by the heuristic are within communities and only rarely are bridges. This means that the difference between maximal similar nodes picked by global and local knowledge mainly only differ in the community they are taken from. Since the local knowledge variant randomly chooses center nodes, the changes are roughly evenly spread across all communities. This is roughly the same effect as choosing node pair similarity globally from communities that all have almost equal similarity rankings among their nodes (caused by the same rewire probability for all communities). Intuitively, the model generates a graph where the most similar nodes of each community have a higher similarity than the second most similar nodes of all communities and so on. The global selection picks the most similar nodes from each community before picking the second most similar nodes from each community and so on. The local knowledge selection will pick center nodes randomly. The local graph will contain the entire community of the node and maybe a few nodes connected by bridges. From this local graph, the most similar node pair will be the the

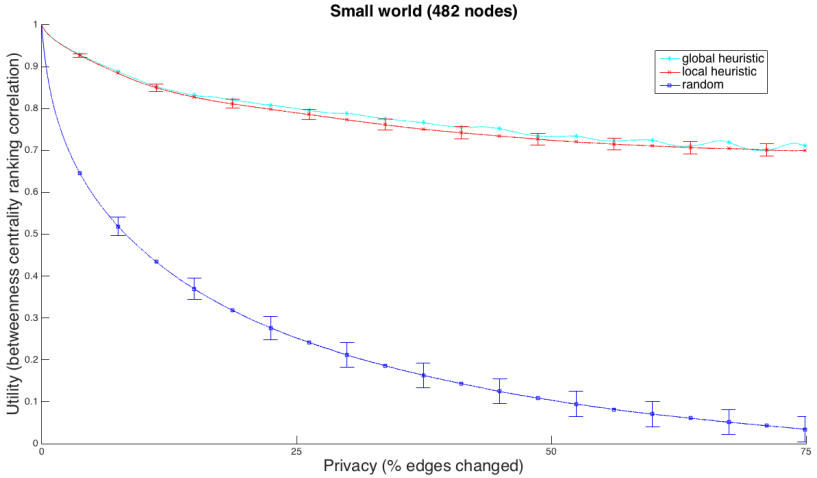


Figure 2.5: *Small-world (482 nodes)*

same as from a global point of view. The random choice of center nodes for each step spreads this effect equally among all available communities. So while some communities might be selected through center nodes multiple times before, another so far unchanged community is selected, only a few "mistakes" are made.

For the smaller graphs in Figures 2.3 and 2.4, the result for the heuristic is not as good as for the larger graphs and a flattening of the curve can be noticed for both at some point. In smaller graphs, the distinction between communities and bridge links becomes more and more difficult, up to the point where the graph consists of only one community. In that case the routing utility and the corresponding ranking is more heavily influenced by the intra-community links (with less explicit structure) than bridges. Isolating only one community, as often caused by small graphs, causes the evaluation metric to only evaluate intra-community structure. Only evaluating one single community has the effect, that similarity among nodes no longer satisfies the assumptions about the graph structure that was originally made when devising the heuristic. I.e. the heuristic tries to keep bridge links, which are by definition not present within a single community. At some point, any existing community structure is fully removed by the algorithm. As a consequence, the curve flattens out, as the routing utility for all nodes is now very close to each other and

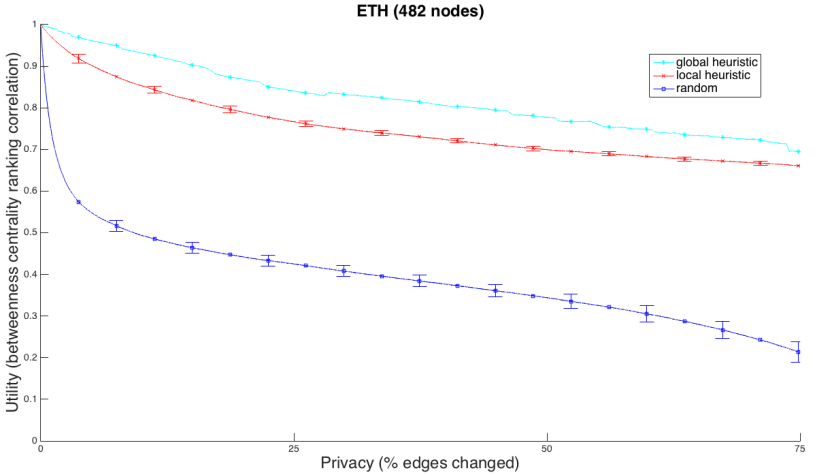


Figure 2.6: *ETH trace (482 nodes)*

further changes by the algorithm do not impact the ranking very much. For the small graphs, the algorithm, as it continues running, eventually accidentally creates a community structure. From that point on, the utility decreases further, as the artificially introduced community structure is different than the original structure, which leads to incorrect edge selection by further running the algorithm.

While heuristic selection does not perform as well for small graphs than for large graphs, it still performs better than random. Furthermore, small graphs can be processed by a greedy selection which makes the use of a heuristic less required. For large graphs, the heuristic performs much better than for small graphs and also than random edge selection.

2.5.3 Evaluation summary

The evaluation results show, that both edge selection variants outperform random edge selection under the assumption of global graph knowledge as well as with only local graph knowledge. For smaller graphs, greedy edge selection performs very well and as the graphs get larger, a heuristic is available that trades of some accuracy with computational load and delivers good results with minimal computational effort over a wide range of edge change

percentages.

2.6 Routing performance

In the previous section we showed, that contact graph privacy can be improved while maintaining important graph metrics. The question we will answer in this section is: *Does routing using a privacy protected graph work?* To answer this question we take a graph-metric based routing algorithm and execute it using our privacy protected graphs.

2.6.1 Routing evaluation methodology

To answer the question how well routing with a privacy protected graph still works, we evaluate the routing algorithm on two traces. Instead of the model based graph of size 20 we use a trace collected at the communication system group at ETH Zurich where the members were carrying mobile devices which used WiFi to detect each other [60]. We also use the Infocom trace already presented in section 2.5.

This evaluation always uses the real contact traces to generate the opportunistic contact events that are presented to the routing algorithm. The decision of the routing algorithm, whether to forward a message or not is based on a contact graph. This contact graph is either the original (real) graph, or a privacy protected version of it. We use the SimBet routing algorithm presented in [20] for our evaluation. Since we do not propose a new routing algorithm, we are not interested in the absolute performance of the routing. We are only interested in the change caused by our proposed privacy protection mechanism as compared to using the original (unprotected) graphs.

In order to quantify the impact of the privacy protection, we use the performance of the routing algorithm using the real graph as a baseline. This allows us to express the impact of the privacy protection as a relative factor to the baseline performance. The metrics we use to evaluate routing performance are *number of delivered messages* and *mean delivery delay*. We generate messages from all nodes to all nodes in regular intervals and measure the time it takes for the messages to arrive at the destination nodes. Messages not delivered after a predefined amount of time are discarded and classified as undelivered.

For each privacy protection type (greedy/heuristic and local/global knowledge) the privacy protection is evaluated from zero to 75 percent and the results are normalized by the performance of the algorithm using the real graph. Thus, values of 1 express that there is no loss in routing performance as compared to using the real graph. Values below one indicate a decrease in routing performance. A value of 0.5 for messages delivered for example means that only half of the messages are delivered when using this privacy protected graph as compared to using the real graph. Values above 1 would indicate an increase in the routing performance metric (either caused by a real improvement or a measurement artifact as will be explained later).

2.6.2 Routing results

Figures 2.7 and 2.9 show the routing evaluation result for the CSG trace and Figures 2.8 and 2.10 show the result for the Infocom trace. All figures show the increasing privacy protection (fraction of changed edges) on the x-axis, the higher the value, the more edges were modified compared to the real graph.

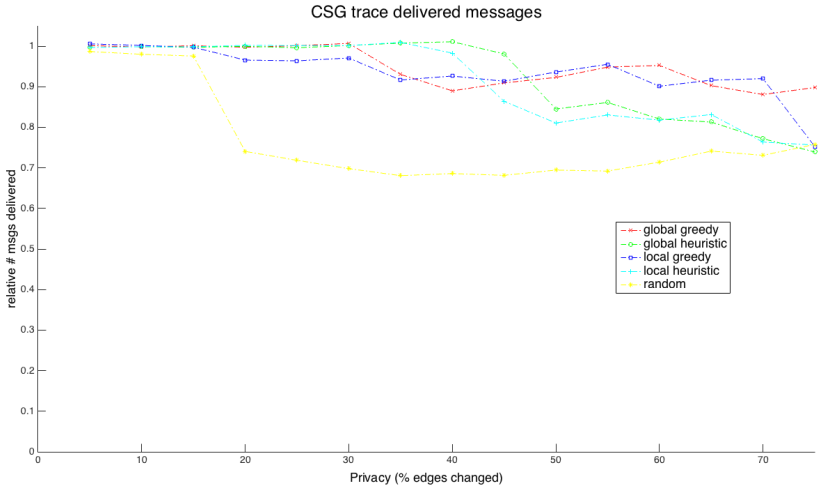


Figure 2.7: *CSG trace # msgs delivered*

Figures 2.7 and 2.8 show the relative number of successfully delivered messages for the privacy protected CSG and Infocom graphs respectively.

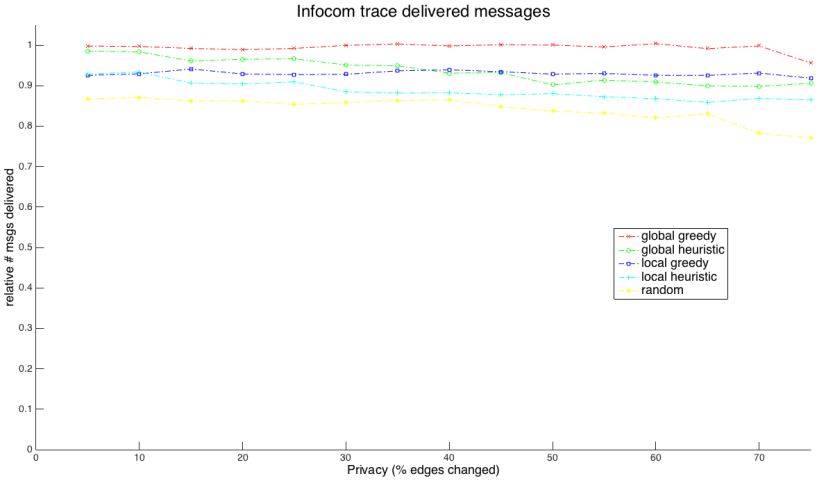


Figure 2.8: *Infocom trace # msgs delivered*

The performance of the randomly modified graphs are presented again as reference. The routing performance on both traces remains high for a large range of edge change fraction values. That means that for the CSG trace, up to 45 percent of the edges can be changed and still more than 90 percent of the original messages are delivered correctly. For the randomly modified graph, this value drops down to 70 percent already after 15 percent of changed edges. In both traces, the performance of the randomly modified graph does not drop to zero. This is expected as the modified graph only influences routing decisions and not the contact events themselves. Whenever a node that carries a message encounters the message's destination node, the message is delivered without considering the routing process. Dense direct contacts in the traces provide for a basic level of message exchanges, that largely does not depend on the routing decisions, regardless of the random changes introduced by our algorithm.

Since the number of successfully delivered messages is well maintained, the next question is, whether it takes longer for the messages to arrive at their destinations. Figures 2.9 and 2.10 show the relative change in the mean delivery delay in our routing evaluation. A relative delay of 1 means, that the messages are delivered within the same time as using the original graph for the routing decisions. A relative delay above 1 indicates, that the messages

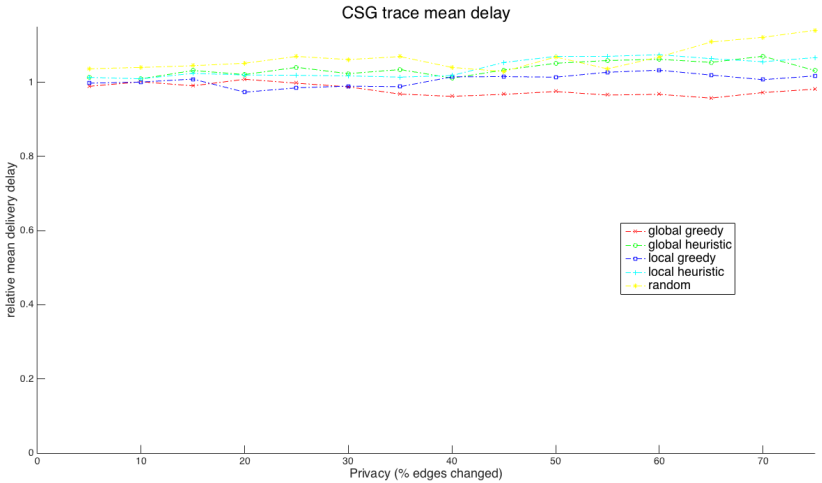


Figure 2.9: *CSG trace mean delivery delay*

take more time for delivery when using the privacy protected graphs and a delay below 1 means, that the messages are delivered faster. For both traces, the delivery delays remain close to 1, which indicates that using our privacy protected graphs does not significantly delay message delivery.

It is important to consider both, the delivery ratio and delay at the same time, as neither delivering all messages over a huge time span (perfect delivery ratio) nor delivering only one message within moments (perfect delay) is a desirable outcome. For example a high delivery rate and a high delay would mean that most messages are delivered, but delivery takes much longer. On the other hand a low delivery rate with a low delay would mean, that only a few messages are delivered, but those are delivered very fast. In our case, the combination of the high relative delivery ratio and about equal delays (values around 1), we can conclude that the average delivery delay remains at a high level compared to the original graph. With increasing privacy protection, the relative delivery ratio decreases and the relative mean delay slightly increases, in other words, fewer messages are delivered and the delivery of those messages takes longer. This is explained by longer paths being cut and thus fewer messages with long delays transmitted. The remaining delivered messages will sometimes be subject to incorrect routing decisions, which then causes longer delays.

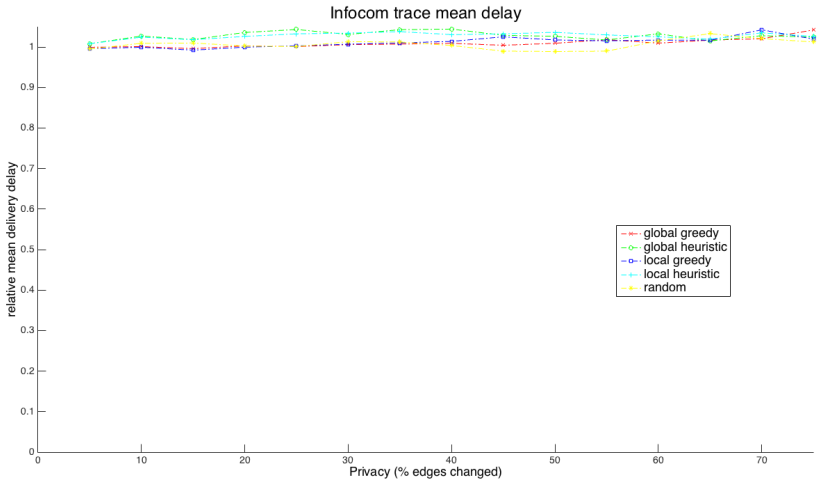


Figure 2.10: *Infocom trace mean delivery delay*

In this section we showed, that our privacy protection approach does not impact routing performance notably. Neither the number of delivered messages, nor the average delivery delay are significantly impacted by using privacy protected graphs for routing decisions.

2.7 Comparison of contact graph obfuscation with the SSNR approach

In this section, we compare statisticulated social network routing as proposed by Parris and Henderson [73] with our approach described in this Chapter. SSNR is designed to protect privacy of individual friend lists attached to each message in opportunistic routing. state of the art opportunistic routing uses contact graphs, which are a more powerful information source and we will show that the SSNR methodology can not be applied to the contact graph based routing algorithms we are addressing in this work. We will first look at the source of social information and the analysis techniques used by the two approaches. Then, we will compare the privacy preserving algorithm and evaluation before looking at the consequences and applicability within state of the art opportunistic networking.

Source and analysis of social information: Parris and Henderson only support explicitly self reported friend lists, whereas our approach supports contact graphs. We do not place any restrictions on how the contact graph is generated. Within SSNR, messages can only be sent to friends and also only friends may forward the message. Thus, contacts with nodes that are not in the friend list are unavailable to forward the message, even though the contact might be able to deliver the message faster than friends. Aggregating information across multiple messages or senders to improve routing performance is not supported.

Our approach protects the privacy sensitive information associated with the social structure encoded by a contact graph and does not have the limitations of SSNR. The contact graph is privacy protected independent of how the graph was generated. In general, our proposal only assumes that the routing input information is given as a graph. It then removes excess information that is not required for the routing decision from the graph, and thus increases privacy of the social information encoded in the graph. In order to achieve this, we apply graph transformation techniques that preserves specific features of the graph. To decide which features we preserve, we survey what features the state of the art opportunistic routing protocols Bubble Rap [52] and SimBet [20] use. They rely on (betweenness) centrality and similarity as main features of the graph. Therefore those are the features we select to preserve in the graph. While this enables the state of the art routing protocols, our approach is not limited to those two features and can be adapted to many other graph features with ease (i.e. node degree).

Privacy protection algorithms comparison: SSNR attaches a friend list to each message, which is privacy protected independently from other messages. The algorithm takes the original friend list as input and adds or removes entries (defined by a system parameter). The authors do not clarify how the nodes to add or to remove are selected. Because the friend list for each message is processed individually, a correlation of protected friend lists of the same sender will reveal the friend list to the attacker. This also means that only local knowledge is taken into account. The authors analysis of the attack resistance against correlation attacks reveals, that an attacker has to intercept 9 messages from the same source before he can reconstruct the real friend list.

In contrast our algorithm rewires edges in the contact graph under the constraint that a given graph feature (e.g. betweenness centrality) has to be preserved. In each step, one edge is rewired in the graph. Thus, the pres-

ence and number of nodes in the graph is decoupled from the social structure, which is represented by the edges in the contact graph. There are no restrictions on which part of the graph is modified, so it operates with local or global information about the contact graph. Thus, the graph could be provided as a global view by some oracle or built up by exchanging local graph knowledge among nodes. The algorithm also maintains a history of modifications to the graph and thus can be executed incrementally whenever the real graph is updated by the external source.

Graph-based comparison One can imagine the friend list in SSNR as a very simple star shaped graph with the sender in the center. The graph only contains links from the sender to the friends and there are no other nodes except the friend nodes. There are no links between friend nodes. This graph will always have a simple star shape. The effect of adding or removing nodes from the friend list is that the star gets or looses one ray. So a node and a link between the center and the node are added (or removed) simultaneously.

Our approach does not modify the nodes in the graph, but only changes edges. As the entire known graph is available to rewire edges, an attacker has to decide whether or not a given edge has been rewired, which can happen throughout the graph and not only centered around a single node. Furthermore, by only preserving the ranking of the nodes regarding a feature, our approach has more liberty in modifying the graph and thus can remove excess social information more effectively. On the one hand, our approach is computationally more expensive than SSNR, but on the other hand our algorithm does not need to be executed for every single message that is sent.

Evaluation summary and conclusion: We evaluate our algorithm first regarding the feature preservation on the graph. With the understanding how rewiring edges impacts the desired feature, we proceed to evaluate our algorithm with an existing routing protocol (SimBet) on available traces without modification. Parris and Henderson directly proceed to evaluate their proposal with their own routing algorithm that is designed to execute their definition of social information based routing. They evaluate routing on available traces, where they are sampling subsets of two trace sets to fulfill the requirement of having enough explicitly reported friend links in the subset to make routing with their approach possible.

We conclude that our approach can be applied to a family of state of the art contact graph based routing algorithms, whereas SSNR has a significantly smaller scope of routing algorithms it supports. SSNR supports routing along predefined node lists, our approach supports routing between any node pair in

the network. Both mechanisms can work with local knowledge, our algorithm additionally supports global knowledge of the contact graph.

2.8 Conclusion

Privacy in opportunistic networks has many aspects. In this chapter, a privacy protection approach for contact graphs of opportunistic networks was presented. Adding and removing edges from the contact graph intelligently allows to maintain the ranking of the nodes regarding a routing metric with few errors. At the same time, it makes it increasingly difficult for an attacker to guess correctly whether an observed link in the contact graph corresponds to a link in the real contact graph. Thus, the social (friendship) structure of the graph is obfuscated while the graph's utility for routing is maintained. This is achieved by careful selection of edges to change, for which we introduce a greedy and a heuristic approach in addition to a random selection of edges. The evaluation showed that the ranking of the nodes regarding a given routing metric can be maintained well for different graphs sizes as well as global and local knowledge. The evaluation of the privacy protected graphs in a routing protocol showed that there is only little impact on the routing performance caused by the privacy protection scheme over a wide range of privacy protection strength. Thus, the privacy protected graphs have excess privacy relevant information removed, while keeping the minimal information set required to allow the chosen routing algorithm to work properly. Future research on this topic could focus on the trade-off of optimizing for certain routing metrics against a general purpose metric that applies to a broader range of routing strategies. This would require to evaluate different routing algorithms based on contact graph metrics as well as more and larger traces. An extended attacker model and the privacy metric can be developed further to cover different attackers and notions of privacy. This would allow to assess, whether the general approach presented here can be successfully applied to an even broader range of privacy threats.

Copyright

2014 IEEE 15th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)

Chapter 3

Privacy preserving social link detection for anonymous opportunistic networks

3.1 Introduction

The previous chapter provided information about how to protect social privacy in the presence of contact-graph-based routing. In detail, the individual social links were obfuscated from the attacker. As such, privacy relevant information about a users social structure is separated and removed from the routing relevant information in the graph. While it is reasonable to remove detailed social link information in cases where it is publicly available, sometimes exact knowledge of social links provide a benefit for other aspects of opportunistic networking.

In most opportunistic solutions in use today, devices each have a single, permanent identifier, which enables routing and other functions, but at the cost of the users' privacy: when an identifier is reused, it was from the same device, and hence the same user, that used that identifier before. If, on the other hand, all users in an opportunistic network are anonymous (and so fully privacy protected), each contact between two nodes is a new contact, since

the privacy protection takes away all information that is required to recognize known nodes. Consequently, recognizing trusted friends is no longer possible, which is a requirement for social-based interaction and cooperation.

Privacy and anonymity are sometimes intertwined concepts. While privacy deals with the protection of personal information about users, anonymity is concerned with protecting a user's identity. As such is is possible to provide anonymity while providing privacy relevant data as well as communicating in a privacy preserving way without being anonymous. The distinction depends on the specific aspect of privacy that is of concern. Especially, privacy protecting the user's identity (providing anonymity) can be seen as one aspect of privacy protection in general. Ideally, if all aspects of privacy are addressed successfully, an attacker can no longer uncover the users identities and as such, full anonymity is achieved. For the purpose of this chapter, we assume such a system that protects all aspects of privacy in opportunistic networks.

The goal of our algorithm is to detect existing social links upon contact, not to find new friends or discover people with similar interests. Instead, we develop an efficient and reliable algorithm that can *detect pre-established social links* with minimal impact on computing and networking resources. Our algorithm requires a previously established relation between the devices, for which we will use the term *social link*, even though that link might have a different meaning in real life (friendship, familiar stranger, ...).

The same goal can be achieved by using privacy-preserving matching (see also section 3.7). However, privacy-preserving matching has a much higher computation and communication overhead and also takes longer than our proposed algorithm. Since social link detection in an anonymous opportunistic network needs to be performed on every contact, when contact times are usually short, our solution is preferable for our use case.

This chapter provides three contributions.

1. We show that existing privacy protection mechanisms directly affect the usability of social links in the opportunistic network that otherwise could be exploited to improve security or performance of the network.

2. We present a mechanism to detect and maintain social connections among the users of the opportunistic network. This allows the network to use additional social-based mechanisms to increase network security or performance. At the same time an eavesdropper can not learn any information about the involved users.

3. We implement our mechanism in a smartphone application and show that its performance is better than privacy-preserving matching, which makes it suitable for deployment in opportunistic networks.

Given these three contributions, this work shows that it is possible to enable generic social interaction even in a fully privacy protected opportunistic networking environment.

After reviewing the privacy challenges in opportunistic networks (Section 3.2), we present our privacy-protected opportunistic networking model as well as the attacker model (Section 3.3). Next, we introduce our algorithm (Section 3.4) and describe possible attacks on it (Section 3.5). We describe our implementation and performance evaluation (Section 3.6) and end with related work (Section 3.7) and conclusions and future work (Section 3.8).

3.2 Privacy challenges for opportunistic networks

Users are at the heart of opportunistic networking: they provide the communication device (typically a smartphone or another personal mobile device carried by the user), the contact opportunities, the bandwidth, and they produce and consume the content. Because these devices are personal, they are used exclusively by their users, and are in close physical proximity to them most of the time. There is thus a strong association between users and their devices.

This close relationship impacts opportunistic networking: how and when devices come into contact is determined by their users, whose movement patterns follow their daily routines and habits. Similarly, the content each user consumes and produces depends on that user's interests (as well as the availability of that content at the time of contact). Device, contacts, and content: these are very powerful sources of information for anyone interested in privacy-relevant data about users. It thus is crucial to know what information needs to be protected and how this protection can be put into place. With growing user sensibility towards privacy, protecting it will become a critical factor in the adoption of opportunistic networking.

The privacy-relevant data that can be revealed about users can be broadly attributed to three categories:

Location consists of the users' geographical locations. This for example includes the place where they live and work as well as the routes they use to commute.

Social information covers all social relationships among users: who is friends with whom, which users meet regularly, for how long and how often, and to which social groups does a user belong.

Content is about users' interests, in this context typically expressed as subscriptions, but also what content is produced by a user.

We concentrate on the social-privacy aspect of opportunistic networking, and specifically on exploiting social links for improving networking services without compromising privacy.

Existing social links are typically independent of any particular application. Therefore, our detection algorithm can be incorporated into the base layer of the opportunistic network. This base layer provides basic application independent opportunistic networking services, such as neighbor detection, subscription matching and content exchange. Privacy relevant information that can be leaked by the base system concerns location and social information about the users. Furthermore, social links, once established, can be used automatically without user intervention which also makes it more convenient for the users. The integration into the base layer makes the social links available to all applications in the opportunistic network.

3.3 System and attacker model

3.3.1 Opportunistic system model

An opportunistic networking system that provides full privacy protection for the user currently does not exist in practice; we therefore assume a system that uses existing protection mechanisms to address all privacy aspects mentioned in section 3.2. This includes anti-location tracking as well as handling existing lower layer identification options such as MAC and IP addresses. Additionally, it would need to use different identifiers for the base layer (neighbor detection, content exchange, etc) and content authoring (generation).

We assume that privacy protection mechanisms can be added to opportunistic networking such that it allows every user to stay anonymous while using the system. For our application we equate privacy protection with full

anonymity; thus, a privacy protected system does not allow permanent or attributable identifiers for data transmission or relay tasks. The effect of full privacy protection is that the user is operating anonymously at any time. One effect, already alluded to in Section 3.1, is that friends can no longer recognize each other. Unfortunately, this also prevents the use of powerful features that build on existing social links to improve networking reliability, performance, security or user experience. Furthermore, this means that it is impossible to "learn" new reliable links based on past contact events and as such automatically extend the social structure (i.e. as could be used for routing).

An example system that would still work in such a perfectly anonymous environment is PodNet [59]. This is an opportunistic content distribution system where publish/subscribe-based user-driven downloads are at the core of the distribution process. Upon contact, nodes offer all available content to the encountered node. The encountered node in turn selects which content it wants and requests and downloads it from its peer. The subscriptions of each device (the users interests) are not explicitly exchanged among the nodes. This way of distributing content does not require routing, and full anonymity would not impact content distribution, as relaying content and generating content are handled independently. Of course, PodNet could benefit from available social links, for example by using them to communicate trusted or reliable content producer identifiers among friends.

3.3.2 Attacker model

Typical attackers can listen to network traffic and try to use the algorithm to learn something about network users, but cannot modify messages in transit (man-in-the-middle) or inject malicious messages that do not follow the algorithm. We do not pretend to be able to defend against attacks from well-funded adversaries of the nation-state type. We therefore consider two different types of attackers.

- The passive eavesdropper who can listen and collect opportunistic networking data. In particular she can collect all protocol messages that are exchanged among nodes. An analysis of these messages should not reveal any information about the social links present in the exchange or the identities of the nodes involved in the exchange.

- The honest but curious attacker who follows the algorithm but tries to gain additional insight by controlling or modifying its own messages.

This attacker has the same capability as the passive eavesdropper but can additionally interact with other nodes in the network. She can modify the content of her own messages to try to extract additional information from the nodes it encounters. Apart from changing message content she still follows the algorithm and all other nodes obey the algorithm as described. The additional capabilities for the attacker lead to control over the social link identifiers that are presented to other nodes as well as "unlimited" retries with every node that is encountered.

Ideally, our algorithm should make it impossible for the attacker to learn anything about the social links in the network. Thus, contacts between two devices would be impossible to attribute to a meeting of specific users. Similarly, relating multiple contacts to the same user(s) is prevented, to avoid detection of friendship or group relationships. In section 3.5, we discuss how well our algorithm achieves these goals.

3.4 Social link detection algorithm for an anonymous environment

In this section, we first introduce *secure pairing*, a secure method to exchange identity information (Section 3.4.1) , before describing our social link detection algorithm proper (Section 3.4.2).

3.4.1 Secure pairing

Recall that the main goal of our mechanism is to detect *existing* social links between nodes that come in contact with each other. This requires that a social link is already present to be detected upon contact. Similar to establishing a friendship in real life, every social link in our opportunistic network needs to be established once.

As our mechanism only detects social links, we consider setting up the social links out of scope of our work. Any social link establishment procedure can be used, as long as it securely generates the following two (optionally three) pieces of information per social link:

- a random social link (friendship) identifier;
- a symmetric key associated with this identifier; and

- optionally, the identity of the user corresponding to the social link identifier

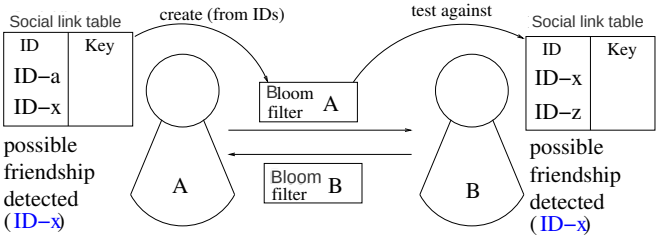
One option to achieve those pieces of information is an implementation of *secure pairing*, which works as follows:

1. Establish a secure association between the two devices. This can be done with an out-of-band channel (e.g. code displayed and entered on both devices).
2. Create a social link identifier (random) and a corresponding symmetric key.
3. Store social link identifier and key for later use (optionally also identity of the corresponding user)

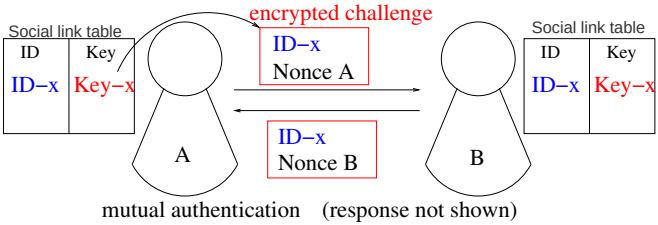
Alternatively, the social links could be bootstrapped using existing social networks (e.g. Facebook, Twitter,...). As this is only required once, it is acceptable if an online connection would be used to set up the social links. Once the social links are set up in that way, they can be used by the system to later recognize friends as described in the next subsection.

3.4.2 Link detection algorithm overview

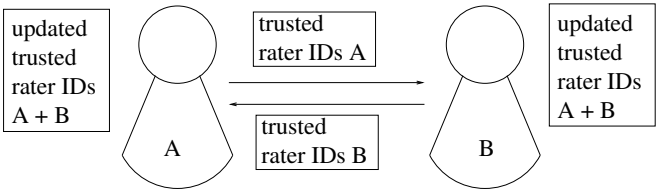
Our social link establishment, detection and maintenance algorithm is illustrated in Figure 3.1 and given as pseudo-code in algorithm 5. Whenever contact opportunities are available, a node picks a contact partner at random, since all are anonymous. Both start the social link detection mechanism by constructing a Bloom filter [11] containing all social link identifiers each device has accumulated by secure pairing. Devices exchange the Bloom filters and locally query the other's Bloom filter against all social link identifiers they own (Figure 3.1 (a)). If there is a match, the devices will challenge each other using an encrypted message that contains the social link identifier and a challenge nonce. This allows the devices to mutually authenticate each other and thus confirm the social link (Figure 3.1 (b)). A new social link identifier is created to replace the existing one (Figure 3.1 (d)). If there is no match in the Bloom filter, there is no social link between the two users. Devices can still continue making the connection available to all running opportunistic applications if they wish. In that case the communication partner is anonymous (and probably untrusted).



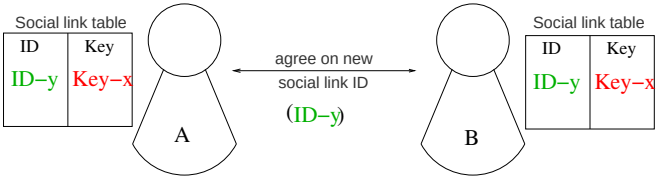
(a) Social link detection step 1 (pick B randomly)



(b) Social link detection step 2



(c) Optional: exchange additional information (encrypted)



(d) Social link maintenance (encrypted)

Figure 3.1: *Social link detection and maintenance.*

Algorithm 5: Social link detection algorithm

```
4.1  connect_to_random_neighbor();
4.2  exchange_bloomfilter();
4.3  query_bloomfilter();
4.4  attempt_authentication();
4.5  if authentication successful then
4.6  |   optional_data_exchange();
4.7  |   mutate_identifier();
4.8  else
4.9  |   ...
4.10 disconnect();
4.11 create_new_bloomfilter();
```

The mechanism can optionally be extended to provide more functionality. After successful authentication the social link key can be used to provision a secure data exchange option to the applications running on the node (Figure 3.1 (c)). We thus have an easy and application-transparent way to encrypt information that flows along the social links. Also, new social link keys can be constructed regularly, in order to prevent a single key to be used for a prolonged period of time. This further increases the security of the algorithm.

We can now go into the details of the algorithm that are relevant for the privacy protection and performance.

3.4.3 Bloom filter details

We use Bloom filters as the main component to achieve privacy for the users. They are used to make the social link identifiers untraceable. If we were using the social link identifiers in their clear form and announce lists for detection, it would be easy to track specific social relations, as always only one identifier will change at any one contact. This is circumvented by using Bloom filters. Using them, it is possible to test whether a specified social link identifier is present in the filter or not. Still it remains impossible to extract the inserted social link identifiers from the filter.

However in its standard form, the Bloom filter creates a “fingerprint” of the currently used social link identifiers, which might already be enough to break user privacy. The hash functions used are known and false positives

need to be kept at a small number for performance reasons (otherwise one could test for all possible social link identifiers without using a Bloom filter). So the Bloom filter itself might be unique enough (even if some small variation is allowed) to track a user over multiple contacts.

In order to prevent user tracking, we use Bloom filters differently. First, the size of the Bloom filter must be the same for all nodes in the system to prevent size-based identification or tracking. Second, to circumvent the filter fingerprinting, we combine three additional measures:

- We use two independent Bloom filters (with different hash functions);
- The set of social link identifiers of a node is randomly distributed across the two different Bloom filters.
- Both Bloom filters are randomly filled with noise up to a defined amount, so that the real fingerprint is hidden in the noise.

Two Bloom filters with independent hash functions are the base to avoid fingerprinting by an attacker. If the same set of hash functions would be used, it would be possible to just add the two Bloom filters and obtain a total Bloom filter containing all social links (and thus the fingerprint). By using more Bloom filters, the number of required hash functions increases as well. An effective way of reducing the number of independent hash functions for a Bloom filter is proposed in [3]. Using this approach, the scheme could if necessary be extended to more than 2 Bloom filters.

The random distribution of the social link identifiers across the Bloom filters for each contact enlarges the set of possible fingerprints. Intuitively, with this measure there are now as many fingerprints for one individual user as there are ways to distribute a node's social link identifiers into two sets (which depends on the number of social links a given user has). As we will show in the next section, if a node has n social links, there are 2^n possible fingerprints.

Filling up each Bloom filter with noise up to a predefined amount further increases the confusion of the attacker, as now all the possible fingerprints are additionally mixed with noise and even harder to detect. Effectively that means that all Bloom filters in the system have the same false positive probability (roughly the same number of bits set).

As a consequence of this construction of the Bloom filter, there is a system-wide maximum number of social links that can be added (for a given false positive rate for the Bloom filter). Exceeding this maximum number will

result in performance degradation as more false positives will appear in the system. Thus, this maximum needs to be large enough so that enough social relations can be accommodated without spending too much time on false positive resolution.

3.4.4 Social link authentication

If the devices match a social link identifier in the received Bloom filter, they need to make sure that it is not a false positive, caused by this particular Bloom filter. To check this, a further step is required that uses the social link identifier and associated key to authenticate the other node. Generally many existing authentication schemes can be used for this purpose.

One option with good performance is a digest authentication scheme similar to the http digest scheme [32]. For this, the first node sends a nonce to the communication partner (to prevent replay attacks). The communication partner uses the nonce, the social link identifier and the key to create a digest answer and sends it back to the initiator. The initiator compares the digest answer with the locally calculated answer. If there is a match, the authentication was successful. In case there are multiple matching social link identifiers, the node sends back multiple digest answers. The number of answers checked by the initiator should be limited in order to prevent authentication digest exhaustion attacks (brute force).

To prevent brute force authentication completely, another authentication scheme that relies on encryption using the social link key would need to be employed. The social link identifier could be encrypted together with the nonce to have a precise authentication check upon decryption of the received answer by the initiator.

While we propose a digest authentication here, this is not a specific requirement for our mechanism and could be replaced by another authentication method as required by the opportunistic networking system.

After an initial social link setup, this mechanism detects existing associations in an anonymous environment. This is achieved by probing for social link identifiers in a Bloom filter that is created from all social links of a user. Proper maintenance of the social link identifiers makes it hard for an attacker to derive useful information from what is exchanged. While the social link detection mechanism comes at a cost, it makes the social structure available for use in the opportunistic environment without compromising privacy.

3.5 Privacy estimation and attacks

3.5.1 Passive attacker

With Bloom filters constructed as described in the previous section, we now look at how difficult it becomes for the attacker to re-identify a given user based on the exchanged Bloom filters. For this analysis, we assume that a given node does not meet any friends at all and no noise is added to the Bloom filters. This is a worst case for our algorithm, as the set of friendship identifiers will remain constant throughout all contacts and is the only case where an exact duplicate of a Bloom filter would be transmitted.

For the passive attacker, we are interested in how many Bloom filter exchanges she has to capture before she is expected to learn anything about the information being exchanged. The best the attacker can hope is for the exact same Bloom filter to be sent in two exchanges. In this case, the attacker knows that the same node is communicating again. As each node sends its own Bloom filter, this does not depend on the node's communication partner. Still, the attacker does not learn about the identity of the node. Given enough such recorded duplicates, the attacker will eventually be able to reconstruct the social link structure within the opportunistic network. This is possible, because the attacker can observe a successful social link authentication, even if it does not know the identities that are involved. Whenever an attacker observes a duplicate in combination with a successful authentication, it learns about one social link. As the attacker continues to observe such events, it can puzzle together the social link structure without knowing the users identities. It then is faced with the task of deanonymizing a known social network, which is a research field of its own. Collecting enough of those events is extremely unlikely, difficult and time consuming but not impossible. This probabilistic protection is the cost of the high performance and small overhead of our algorithm. In order to determine how likely such a privacy breaching event is, we look at the probability for the attacker to capture such an event under the worst case assumption for our algorithm as described above. The notation we use for this analysis is as follows

sl	number of established social links
N	total number of Bloom filter mutations
n	expected number of eavesdrops for attacker

The less variation in the construction of the Bloom filters, the higher the chance for the attacker to capture a duplicate. The worst case for our algo-

rithm additionally distributes the existing social links evenly over both Bloom filters. Thus, both filters contain the same number of elements. In this case there are sl IDs to be distributed into two sets of equal size $sl/2$. Thus, there are $N = \binom{sl}{\frac{sl}{2}}$ ways to distribute sl IDs among the two filters.

For a given number of N possible Bloom filter variations, we will see on the average about $n = \sqrt{(2 \ln 2)N}$ exchanges before a specific variation of the filter is exchanged twice. We compute new Bloom filters for each exchange, therefore the problem of duplicates is the same problem as the one known as the birthday problem. In the generalized version of this problem, there are N possible birthdays and we compute $p(n)$, the probability that out of n randomly selected people, at least two have identical birthdays. We then ask, for what n is $p(n) > 0.5$.

Let us first compute $q(n)$, the probability that all n birthdays are different. Assuming that birthdays are uniformly distributed and that the draws are independent, we have:

$$q(n) = \prod_{k=0}^{n-1} \left(1 - \frac{k}{N}\right). \tag{3.1}$$

Using the Taylor approximation $e^{-x} \approx 1 - x$ for $x \ll 1$, we get

$$q(n) \approx \prod_{k=0}^{n-1} e^{-k/N} = e^{-n(n-1)/2N} \approx e^{-n^2/2N}. \tag{3.2}$$

Now we want to know for which n we expect two or more people to *share* a birthday, in other words, for which n we have $1 - q(n) > 0.5$:

$$\begin{aligned} 1 - q(n) > 0.5 &\Leftrightarrow 1 - e^{-n^2/2N} > 0.5 \\ &\Leftrightarrow e^{-n^2/2N} < 0.5 \\ &\Leftrightarrow -n^2 < -2N \ln 2 \\ &\Leftrightarrow n > \sqrt{(2 \ln 2)N}. \end{aligned}$$

If we relax the restriction of the equal distribution of social links among the two Bloom filters, the number of possible combinations increases slightly, as does the required number of eavesdrops for the attacker, since now there are $N = 2^{sl}$ possible combinations and the expected number of exchanges would therefore be on the order of $\frac{2^{sl}}{2}$.

Keep in mind that this is a worst case analysis for our algorithm. In practice, the social link identifiers inserted into the Bloom filter will change as social link identifiers change each time a social link is met and authenticated. Furthermore, the addition of noise to the Bloom filter (non existing social link identifiers) is not considered in this analysis. The addition of those two additional protection mechanisms further increases the difficulty for an attacker to even re-identify one single user for one time.

For an example analysis we assume 400 friendships established and a Bloom filter false positive ration of 5% for each individual (half) Bloom filter. The Bloom filter information in this example is calculated as in [11]. In a conservative approach (with a half-half split), each partial Bloom filter has to accommodate 200 friendship identifiers which leads to a size of 1248 bits per filter with the optimal number of 5 hash functions to maintain the chosen false positive rate. In that there are $\binom{400}{200} \approx 10^{119}$ variants of splitting the friendship identifiers over the two Bloom filters. The attacker would have to record an expected number of $\sqrt{2 \ln 2 \binom{400}{200}} \approx 3.7 \cdot 10^{59}$ friendship detections to see a duplicate Bloom filter and thus re-identify a given node. If we relax the restriction of the half-half split, then the number of possible combinations would go up to $2^{400/2} \approx 1.6 \cdot 10^{60}$.

The general number of required eavesdrops is exponentially proportional to the number of social links one user has. Already with a relatively low number of social links the attacker has to collect a large number of exchanges in order to re-identify one node. Adding noise to the Bloom filter further increases this number significantly, as this virtually increases the number of friendship identifiers present in the filter. This additional increase comes at the cost of falsely detecting friendships that do not exist.

3.5.2 Active attacker

While a passive attacker can only hope to see a duplicate Bloom filter, the honest but curious attacker has some more options at hand:

- Engage other nodes in additional exchanges, and to
- control the social link identifiers within its Bloom filter.

As all nodes are acting anonymously, an active attacker can connect to a victim node as often as it wants to and pose as a new contact. By engaging nodes in additional social link detection exchanges, the attacker is able to

speed up the collection of transmitted Bloom filters. Doing this, she increases her chances of seeing a duplicate Bloom filter and thus re-identifying a user. Preventing this is helped by the number of expected Bloom filters that needs to be captured before a duplicate is seen, as well as the mutating Bloom filter as other nodes meet and authenticate social links.

Controlling the social link identifiers contained in its Bloom filter allows the attacker to probe for specific friendship identifiers. This way, the attacker can test whether or not its target has a given specific social link identifier in its list. Probing is not limited to one single identifier, but also works for a group of social link identifiers. This can also be achieved (though with a little less precision) by the passive attacker testing the captured Bloom filter against the given set of identifiers. Since the social link identifiers change after each use and are generated randomly, it is impossible to predict them and thus trace a given social link over multiple contacts.

Another active attack vector is the Sybill attack [22] where the attacker generates a large number of non-existing nodes. Since social links are explicitly established by the involved users directly, additional (virtual) nodes can not authenticate with real nodes. Even if the attacker has somehow created a social link with a single user, there is no transitivity of trust and thus the Sybill nodes do not benefit from it. It is however possible to mount a form of denial of service attack by providing a large number of identities for the detection phase. As the detection has to select one node at random, this can effectively starve the real social contacts.

3.5.3 Bloom filter space

It is possible to relax the attackers goal from exactly re-identifying a user to a probabilistic detection of “near-duplicate” Bloom filters under the worst case assumption. Depending on the confidence required by the attacker to classify two Bloom filters as “near-duplicate” the number of combinations to distribute the set of social link identifiers over the two Bloom filters needs to be reduced further. Our algorithm can easily be extended to integrate a “near-duplicate” detection or prevention before sending out the Bloom filter (e.g. check for a minimal amount of bit differences in the individual Bloom filters). However, this would reduce the number of total combinations and thus slightly increase the chance for the attacker to see an exact duplicate.

3.5.4 Brute force attack on the Bloom filter

Another attack vector is to brute force the Bloom filter in order to extract the list of friendship identifiers. Doing so successfully would allow an attacker to later query an intercepted Bloom filter for an exact list of friendship identifiers, thus re-identifying a node with a high probability. This subsection describes the attack and puts it into context with our system and attacker model.

In general, a brute force attack on a Bloom filter starts with intercepting a first filter and testing it against the entire space of possibly inserted elements. This leaves the attacker with a subset of candidate elements, with a size related to the false positive rate of the Bloom filter. With every consecutively intercepted filter, the attacker can further reduce the candidate set, until it is left with a set of elements that are present in the filter with a high probability.

Initially, we focus on the difficulty for the attacker to brute force the Bloom filter as is it used in our algorithm. The first step is to test an eavesdropped Bloom filter against the entire friendship identifier space. This will yield a candidate set in the size of approximately 2 times the false positive rate of the identifier space (as we are using two distinct Bloom filters to represent the entire set). With every additionally intercepted Bloom filter, the attacker can reduce the friendship identifier candidates to approximately 2 times the false positive rate of the current candidate set, until the candidate set becomes constant. Consequently, there are two parameters that control the amount of work an attacker has to put into the attack: the size of the friendship identifier space and the Bloom filter false positive rate. Both are system parameters that can be chosen to meet a certain attack resistance. The higher the chosen values, the more difficult it becomes for an attacker to brute force the Bloom filter, both in terms of required computing power as well as the number of required Bloom filter eavesdrops. The number of required eavesdrops starts at approximately 10 for a friendship identifier space of 32 bit and a false positive rate of 1 percent and increases to 39 with a friendship identifier space of 128 bit and a false positive rate of 5 percent. Those numbers are lower bounds, as added noise to the Bloom filter is not taken into account.

Additionally, considering the properties of our algorithm and the system model, there are further obstacles the attacker has to overcome. First, friendship identifiers are changed whenever the two nodes that share this friendship meet. Therefore, the attacker has to intercept enough Bloom filter variants from a single node before this node meets one more more (depending on the attackers desired accuracy) of his friends. To reconstruct an identifier after it

was changed by the nodes, the attacker has to start the attack from scratch, as the identifiers are generated randomly. Second, our algorithm operates in an anonymous system model. Under anonymity, it becomes much more difficult or impossible for an attacker to collect multiple Bloom filter variants from the same user, because it is unable to attribute multiple eavesdropped Bloom filters to one single node. If an active attacker can use its abilities to obtain enough Bloom filter variants from one single node (e.g. by fast and repeatedly acting as new contact) depends on the properties of the anonymity system and is thus out of scope of this work. However, the difficulty of an active attack can be further increased by the size of the friendship identifier space (which increases the computational load for the attacker). Furthermore, an extension to our algorithm, where loosely synchronized nodes negotiate a set of friendship identifiers instead of only one and rotate through this set can restrict an attackers success to a limited window of opportunity. To sum it up, our algorithm has two parameters that can be chosen based on the properties of the anonymous opportunistic network environment. Doing so, it becomes impossible for a passive attacker to collect enough Bloom filter variants from one single node for a successful attack.

3.5.5 Attacks that bypass the algorithm

There are attacks that can be mounted against a fully privacy protected opportunistic networks that do not run our social link detection algorithm. Such attacks can try to exploit the daily routines of users by combining location and timing to recognize users (as they tend to be at the same locations at the same times often). Our algorithm does not protect against such kinds of attacks, but also does not add information that makes those attacks easier. Counter measures against such attacks will in most cases be deployed in parallel to our algorithm without interfering with it.

3.5.6 Inference attack

Listening to traffic, an eavesdropper could infer whether there is an existing friendship or not by the presence of friendship maintenance messages, which are only sent among friends. Either this is not considered a privacy problem, as both partners are anonymous and the exchange is encrypted, or dummy messages could be sent by non friends to masque this information (in a cooperative environment). Additionally, application specific data can

be exchanged once a valid friendship is detected. This exchange is protected by the friendship key and provides a generic protected channel for applications to exploit the existing social structure in the contacts. The mechanism introduces additional overhead to the system. First, the selection of communication partners has to happen randomly. Depending on the number of contacts and friendships it may take some time until a connection to a friend is established. However, the existing social structure works in favor of connecting to friends, as friends are usually met more frequently and thus have a higher chance of connecting. Second, the additional creation and exchange of protocol messages consumes time and resources.

3.5.7 Privacy estimation summary

Implementing our algorithm provides probabilistic privacy protection while reliably identifying existing social links. However small, attacker gains the chance to re-identify a user in two exchanges when our algorithm is used. We showed, that the probabilities to re-identify a user are extremely small for an attacker and thus we believe the benefit of our algorithm outweighs the potential privacy loss.

3.6 Performance

In this section we investigate the load that our algorithm places on the system. We first look at the basic algorithm load and then at the additional overhead that may happen due to probabilistic collisions since our algorithm generates identifiers randomly and the use of hashing.

3.6.1 Base load

There are four main elements in the algorithm that make up the base load:

- creating the Bloom filters,
- exchanging the Bloom filters,
- querying the Bloom filters,
- authenticating a social link.

Preparing the Bloom filter involves distributing the existing social link identifiers into two sets and calculating the hash values for all identifiers. Creating the two Bloom filters is dominated by hashing each identifier with a given number of hash functions. Preparing the Bloom filter can be done before initiating an algorithm run. The new filter can be prepared right after an algorithm run is finished. Thus, it does not add to the communication / detection run time in practice. We implemented the Bloom filter creation for Android smartphones and measured the creation times for different Bloom filter properties and number of friendships on Google Nexus 4 phones. The results can be seen in figure 3.2. The plot shows the mean and standard deviation for 1000 runs. The time to create the Bloom filter increases linearly with the number of friendship identifiers inserted. We ran the same code on a Samsung Galaxy S III which interestingly did the calculation almost twice as fast as the (newer) Nexus 4.

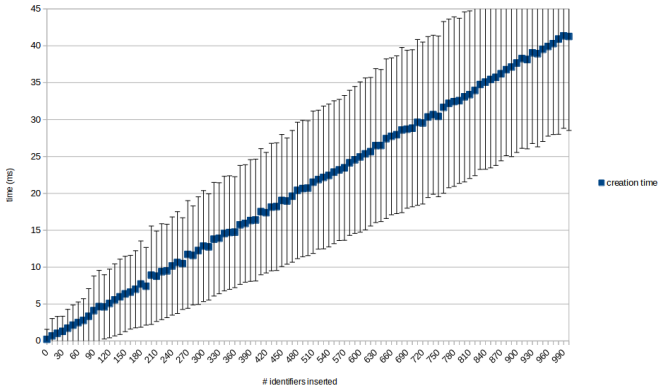


Figure 3.2: *Bloom filter creation times on Nexus 4 (false positive prob.: 1%)*

If the Bloom filter is available, the algorithm just needs to embed it into a network message and transfer it. So the load generated here is network data transmission. As the Bloom filter size depends on the number of expected inserts and the desired false positive rate, the data volume also depends on those factors. Generally the Bloom filter size for practical applications will be one thousand to a few thousand bit, so it will fit into one single network packet. Querying the Bloom filter depends on the number of social link identifiers that need to be checked. Since querying can be stopped for each identifier as

soon as one hash bit is not set in the filter, the query operation is faster than the creation. The load of the authentication step depends on the specific authentication mechanism employed. In the case of the proposed simple digest based authentication, a random nonce value is transmitted, one hash calculated and the result transmitted back and compared to a locally generated hash value.

We measured the algorithm run time with our Android application. We set a Bloom filter with 1000 insertions, a false positive probability of 1 percent and one successful social link authentication was performed using our proposed digest authentication. On a Nexus 4 phone the entire run took about 55 ms with a standard deviation of 35ms (caused by the network) over 200 measurements.

Most related work in section 3.7 does not include performance evaluation. Only the authors of FindU [61] provide performance simulation for smartphones. While the performance of their algorithm depends on multiple parameters, their simulated protocol run time is ranging between .5 seconds in the best case and 5 seconds in the worst case, for one specific scheme up to 50 seconds. FindU and our algorithm are not aimed at exactly the same problem, but the run times for FindU give an idea about expected run times if privacy preserving matching would be used to achieve the goal of our algorithm. The expected run times would be 10 to 100 times longer.

3.6.2 False positives

Apart from the base load of generating and testing the Bloom filter for known friendship identifiers, false positives (detecting nonexistent friendships) are the second possible performance aspect we investigate. In short, a false positive makes a node temporarily think that it is connected to a friend when it is not. Every false positive incurs an additional overhead due to the authentication step that is performed and which fails in that case. A false positive can only occur during one specific step of our algorithm. This step is, when a node checks the received friendship identifier list that is represented as a Bloom filter against its own local list of established friendships. If the local node detects a matching friendship identifier in the received and local list that is in reality not related to one of its pre established friendships, we call this a false positive. Our mechanism has two possible independent sources of false positives, the friendship identifier generation and the Bloom filter query.

Friendship identifier generation

As friendship identifiers are generated randomly, it is possible that two independent pairs of nodes generate the same friendship identifier for their own distinct friendship relations. In this case, a matching friendship identifier is detected during the Bloom filter query process. Consequently, the algorithm will go to the friendship verification step (challenge - response) where the corresponding keys are different for each node, the challenge - response authentication fails and the false positive is detected. However, this creates an additional challenge response exchange for each false positive caused by a friendship ID duplicate. The expected number of this type of false positives depends on the size of the space that the friendship identifiers are chosen from as well as the total number of friendships in the system. As duplicate friendship identifiers on nodes that are never meet will never cause a false positive, the latter can be reduced to the total number of friendships within all contacts of a node. The larger the friendship identifier space, the fewer false positives are expected. The expected duplicate (and thus false positive) probability for two nodes meeting is shown in Figure 3.3 for 16, 24 and 32 bit friendship identifiers and nodes having between 1 and 1000 independent friendships each. It is assumed that friendship identifiers are drawn uniformly random from the friendship identifier space.

Bloom filter

The Bloom filter guarantees no false negatives, so if we do not find one of our friendship identifiers in the filter, we definitively are not connected to a friend. False positives result when the Bloom filter query results in a match where there is no matching entry in the friendship identifier list that was inserted into the filter. Typically, a false positive query result will cause the involved nodes to proceed to the authentication step, which consequently fails. Furthermore, false positives can cause a node pair to detect more than one shared friendship (which does not make sense, as two users can either have one social link or none). In this case, the authentication step is performed each detected friendship. The other party will try to decrypt all messages with the keys that are associated with the friendship identifiers it found itself (which might be one or multiple). Correct decryption can be verified since the friendship identifier is contained in the encrypted message. The false positive rate of a Bloom filter is controlled by the size of the filter, the number of hash functions used in the generation process and the expected number of elements that are

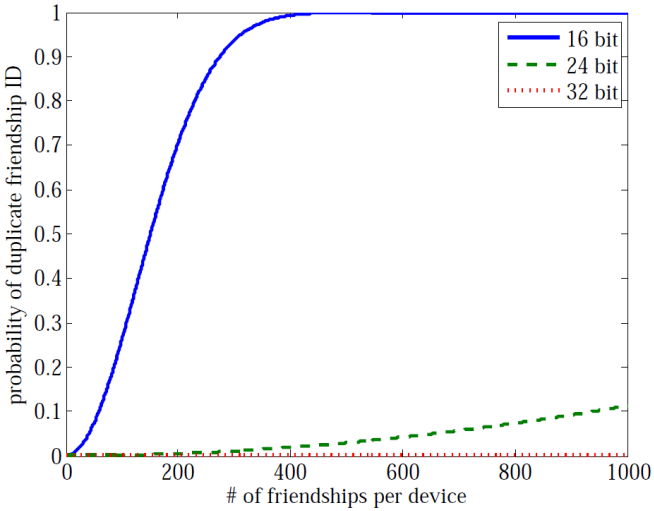


Figure 3.3: *Duplicate friendship ID probability in node meeting for different ID sizes*

inserted into the filter. Therefore, the Bloom filter false positive rate can be controlled in the system.

3.6.3 Application integration

In addition to evaluating the algorithm implemented as an Android application, we also integrated the approach with a full-fledged opportunistic networking application. The Android implementation of the PodNet publish-subscribe content distribution served as a basis for the integration of our algorithm. While it does not operate in a perfectly privacy protected environment, it is a complete content distribution application and allows for all steps of our proposed mechanism to be executed in the context of a real application. The privacy preserving friendship detection is implemented as a separate Android service that is called whenever neighbors are in range. The integration with the PodNet application has a slightly higher overhead than the stand alone implementation we used to investigate the overhead caused by our algorithm alone. This is caused by interaction with the PodNet application, which han-

dles neighbor detection as well as all communication set up (e.g. establish a TCP connection). We did not include the connection establishment in our core performance, as this has to be done anyway to communicate in the opportunistic network. A further investigation of the additional overhead revealed, that it is mainly caused by the Android Intent inter process communication that was used. Using another IPC mechanism available in Android (Binder) results in a significant drop of the additional overhead, to an almost negligible value. This is confirmed by measurements performed by [51], who compared various Android IPC mechanism properties.

3.7 Related work

In this section we look at related work in privacy protection that can be applied in the context of opportunistic networking.

Social privacy in opportunistic networks has mainly been studied under the premise of routing information as in [72]. Privacy of social group membership has been investigated by [9]. The problem of finding common information among two nodes in a privacy protected way is achieved by research in the domain of privacy preserving matching or privacy preserving set intersection [34]. Freedman et al. propose a mechanism where only the "client" learns the result of the computation (but not the server), which in our case would require a client-server role negotiation beforehand to make it applicable. Privacy preserving matching and private set intersection (also private cardinality set intersection) are crypto-based techniques to calculate if two (or more) parties share a common piece of information, without revealing their respective entire information set. This can potentially solve our task to find out, whether two communicating users have a social link identifier in common, without revealing their entire set of social link identifiers to each other. With FindU [102] the authors present an efficient privacy preserving match making protocol. However, their goal is to find the best matching user among a group of users, and thus the protocol does not work for an encounter of only 2 nodes. Privacy preserving friend discovery is also done in [21]. The authors provide a solution for finding social proximity (detecting new friends) given the social coordinates of the two users and not to detect existing friends. It also relies on a trusted central server that computes the social coordinates for each user which is later used offline and needs to be updated regularly. Privacy preserving matching has a high computational demand that makes it unsuitable for a situation as ours, where social link detection has

to be done frequently at the beginning of each data exchange. Furthermore, some schemes rely on multiple parties (nodes) being involved in the calculations, which is a requirement that can not be satisfied in all contact scenarios in opportunistic networks.

For E-SmallTalker [106], Yang et al. use iterative Bloom filters for common interest detection. They use Bloom filter as a compression tool to fit the interest list into size limited Bluetooth service discovery packets. There is also a proposal for cryptographically secure Bloom filters by Nojima and Kadobayashi [67]. While it basically satisfies the requirements to be used in our mechanism, its computational complexity is too high for implementation on smartphones, which are the typical opportunistic networking devices.

3.8 Conclusion

A major drawback of fully anonymous (privacy protected) opportunistic networking is the lack of any exploitable social structure. In this chapter we present a mechanism that allows to make use of social links in an otherwise perfectly privacy protected network. Thus, the social structure inherent to an opportunistic network becomes available for reliability and security mechanisms while maintaining the privacy and anonymity of the users. The social link detection mechanism provides a generic, application independent way of recognizing social links. It makes those links available for other applications for encrypted and authenticated data transfer, so it can be incorporated in an "opportunistic base-layer" as application independent service.

Our goal to design an application independent algorithm may imply some limitations in certain scenarios which we briefly discuss now. The maximum number of friendships any node may create is the same for all nodes and a system wide parameter. This may be a limiting factor in special purpose applications outside of traditional opportunistic networking. We believe, that this does not present a limitation for general purpose use in opportunistic networking, as the number of friends a user can be in contact with physically, is naturally limited by everyday life. Since the maximum number of friendships is not bound, but only needs to be set to a specific value, this value can be chosen high enough to accommodate most users behavior. Furthermore, nothing prevents a social link to be removed at any point in time (e.g. deleted from the list). Generally, a deleted friendship can be replaced by a new one. Depending on the requirements of a special purpose scenario, various friendship replacement strategies could be implemented whenever the maximum num-

ber of friendships is reached (e.g. replacing the least recently used friendships first).

In scenarios, where available computing power is lower than that of smartphones, which are typically used for opportunistic networking, or where contact times are generally much shorter than in the opportunistic case, further performance tuning of our algorithm could be required. One option that reduces the number of exchanged messages is, that only one node sends its list of friendship identifiers to the other instead of a mutual exchange. Furthermore, the maximum number of friendships allowed can be set to a low value, which reduces both, bloom filter size and creation time.

Due to the application independent approach, the presented attacker model only operates with information that is produced by our algorithm. A more realistic attacker model would include further background knowledge, which makes an attack more effective. We believe, that our attacker model serves as a good base for further attack scenarios that may take scenario specific background knowledge into account.

Copyright

Part II

The influence of smartphone type devices on opportunistic contact properties

Chapter 4

Smartphone WiFi design influence on opportunistic contacts

The previous part has dealt with keeping opportunistic contacts and derived information privacy protected. If contacts get in the focus of investigation, this typically happens to extract statistical data about contact timing and frequency. During a CTI funded knowledge transfer project to a startup, that was conducted during this thesis, a different view on contacts in practical opportunistic networking was adopted. A contact is typically understood as a time period, in which two mobile nodes are within each others radio propagation range. In most cases, the propagation range is modeled as a disc centered around a node, where the node itself does not occupy any space. This part of the work questions this typical view and investigates the reality behind smartphone based opportunistic contacts.

4.1 Introduction

Opportunistic networks are envisioned to provide communication in many different scenarios. In all of those scenarios, such as emergency communication, cellular offloading or mobile social networking, users are a core element of the network. The user is assumed to carry a mobile device that is capable

of establishing wireless connections to other users carrying a similar device in the vicinity. As such it is not surprising, that the characteristics of opportunistic contacts are attributed to the users' properties, such as mobility, daily routines and social connections. Therefore, mobility models and contact timing properties (e.g. inter contact times, frequencies) have contributed to expanding our understanding of opportunistic networking. This again has lead to numerous proposals and algorithms that improve various aspects of opportunistic networks, for example routing, trust or security. The mobile device carried by the user is regarded as a mere enabler for opportunistic technology. If at all, the maximum communication range is considered a limiting factor for the contacts. Thus, rough range estimations are typically used, sometimes two different communication technologies (e.g. Bluetooth and WiFi) are used in evaluations, only distinguished by their maximum communication range. The modeling and simulation techniques that are used often simplify mobile device properties, which leads to significant deviations from the real communication properties of mobile devices.

Yet, the potential of opportunistic networking lies in extending the reach of the Internet to rural and remote areas [43]. Facebook with the Internet.org project and Google with its Project Loon are now rapidly developing Internet connectivity for the still disconnected 4 billion human beings. They plan to build novel infrastructures no longer relying on classical wireless cell infrastructure, which is very costly to deploy and maintain in remote areas. Instead, balloons (at 32 km of altitude) or drones (at 18 to 27 km) will build a floating mesh infrastructure supporting fixed ground WiFi AP stations or cellular connectivity. As such stations will be costly to deploy, they will be located at strategic points (e.g. building rooftops, villages). Connecting end users with those stations is where opportunistic networks will play a key role as the "last mile connectivity" solution to extend the reach of such stations beyond their limited range.

The second and often neglected core element supporting opportunistic networks is smartphones. Already smartphones in the 30$ range are reality [84] are reaching emerging markets. While the potential for opportunistic networking has already been demonstrated through the N4C project [29] or apps such as Twimight [27], Uepaa [97] and Firechat [68], there is only little evidence of the large-scale potential of opportunistic networking based on smartphones. In practice, the contact opportunities – the building blocks of opportunistic networks – depend to a large extent on the characteristics of the mobile devices, typically smartphones, that are carried around. Up to now,

no study has ever considered the radio characteristics of current smartphones acting as opportunistic nodes and the radio impact of human beings carrying these devices. Smartphones have their own characteristics (i.e. output power, antenna gain) and the human body heavily impacts transmissions (e.g. by absorbing a part of the emitted signal). The properties of smartphones have not received the same amount of attention as the "user" part of opportunistic networking. We noticed while running simulations with NS-3 a significant discrepancy between the embedded models and preliminary tests with smartphones, which led us to this study aiming at fully characterizing smartphones' WiFi performance for opportunistic networking.

In this chapter, we hence strive to answer a simple question: Is WiFi-based opportunistic networking feasible with current smartphones? If yes, what is the performance of WiFi-based opportunistic radio link with those devices in a real-world environment? We start by revisiting the link budget between two smartphones in Section 4.2. Next, we detail the internal characteristics of different smartphones (antenna type and gain, WiFi chip set and output power) in Section 4.3. After a reminder on propagation models in Section 4.4, we report about our line of sight (LoS) outdoor field measurements from which we derive the best path loss model in Section 4.5. We evaluate the impact of the body attenuation model in Section 4.6, assess the empirical maximum LoS range in Section 4.7 and eventually derive the opportunistic WiFi link capacity of crossing pedestrians. Eventually, we survey related work in Section 4.8 and discuss open points and future work in Section 4.9.

The main contributions and findings of this chapter are:

- A fully characterized smartphone link budget from smartphones' output power to their reception threshold.
- A calibrated Two-ray Ground model and a body model derived empirically.
- Line-of-sight range measurements with different smartphones, evaluating practical communication range (up to 400m) and good-put of two crossing pedestrians of at least 143 MB.

4.2 Smartphone link budget

The link budget generally reflects all gains and losses that RF signals are subjected to while traveling from the transmitter to the receiver. Our methodolog-

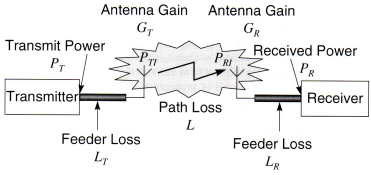


Figure 4.1: *Radio communication chain.*

ical teardown will follow the sequence of the individual items of the radio link budget. This gives us a step-by-step understanding of the smartphone WiFi properties for opportunistic networking. We start by detailing the WiFi link budget between two smartphones. Link budgeting usually aims at setting the transmitter output power P_T such that the received power P_R at the receiver allows establishing a communication channel. In our case, we have no degree of freedom since we cannot control any WiFi parameter on smartphones. Besides, compared to traditional link budgeting, we must account for the impact of the carrier's body since smartphones are usually carried in pockets or used in close proximity to the body.

A link budget accounts for all the gains and losses from the transmitter, through the medium to the receiver as illustrated in Figure 4.1.

A radio signal sent from one smartphone to another is emitted by the senders WiFi chip set with power P_T. A part of the emitted power L_T is lost in the feeder line that connects the chip set to the antenna built into the sender's smartphone. The antenna characteristics G_T determine the emitted power of the radio signal. The emitted signal power decreases by a certain amount L depending on the distance to the receiver's phone antenna as well as due to any obstacles that it potentially travels through, such as the body of the phone user B_T. All the factors except the path loss have to be considered for the receiver as well as for the sender until the final signal strength at the receiver chip set can be determined. The link budget equation is given by:

$$P_R = P_T - L_T + G_T - B_T - L - B_R + G_R - L_R \tag{4.1}$$

where:
(emitted and received power is given as absolute value in dBm,
all other factors are relative changes to this given in dB (or dBd for Antennas))
P_R = received power (dBm)

P_T = transmitter output power (dBm)
L_T = transmitter losses (cable, feeder/connector, ...) (dB)
G_T = transmitter antenna gain (dBd)
B_T = transmitter body attenuation (dB)
L = free space loss or path loss (dB)
B_R = receiver body attenuation (dB)
G_R = receiver antenna gain (dBd)
L_R = receiver losses (cable, feeder/connector, ...) (dB)

The difference between the received signal power, P_R, and the sensitivity of the receiver is referred to as the link margin. The sensitivity of the receiver depends on the signal bandwidth, the type of modulation and the noise level. The WiFi 802.11b/g thermal noise power (or floor) with a channel bandwidth Δf of 20 MHz is:

$$P_{dBm} = 10\ log_{10}(k_B T \Delta f \times 1000) \tag{4.2}$$

where $k_B =$ is Boltzmann constant ($1.3806504 \times 10^{-23} J/K$ (Joule/Kelvin)) , Δf is the bandwidth in Hz and T the room temperature t in Kelvins ($T = 273.15 + t$ in Celsius). At $20°C$ the thermal noise power is -100.9 dBm and at -$5°C$ it is - 101.3 dBm and will hence be roughly the same whatever temperature (winter vs. summer) Thermal noise can be approximated by $-174\ dBm + 10\ log_{10}(\Delta f)$ resulting in an average noise power of 101 dBm. The signal to noise ratio (SNR) must be at least 4 dB in order to achieve a bit-error rate (BER) of 10^{-2} required for the lowest rate of 802.11b (DSSS - DBPSK). This results in a reception threshold of -97 dBm.

In the remainder of this chapter, we will investigate all the above link budget parameters through publicly available data, measurements and provide models whenever possible.

4.3 Inside smartphones

Based on publicly available data (e.g. FCC), we report about the position and type of the WiFi antenna and their gains. We then provide a few WiFi chip set characteristics and the radiation patterns of smartphones.

4.3.1 Antenna characteristics

All recent smartphones are equipped with Planar Inverted-F antennas (PiFa) that belong to the family of patch antennas. A PiFa antenna is resonant at a quarter-wavelength thus reducing the required space needed on the phone. With PiFas the entire ground plane that supports the circuit board and touch screen (i.e. the entire phone) makes up the antenna. Hence, the bigger the phone, the better. Figure 4.2 shows the layout of the different antennas as observed on most smartphones (rear view). The GPS antenna is located on top for maximum reception and there are two cell antennas: one RX only at the top and one at the bottom for TX and RX, so that the EM exposure is reduced at the ear level. The WiFi PiFa is always located on the rear's right side and spans almost the whole phone length as shown by the dashed line. Antenna gains reported by the FCC vary from 1.1 dBi for the HTC Nexus One to -2.5 dBi on the the HTC One X and -1.5 dBi on the iPhone 4S.

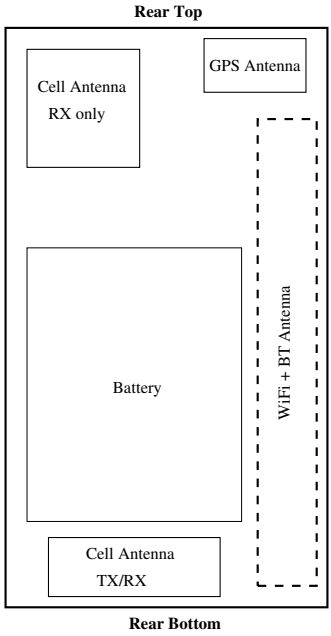


Figure 4.2: *General layout of the different antennas. The dashed rectangle indicates the typical position for the WiFi PiFa.*

4.3.2 WiFi chip-sets characteristics and emitted power

Table 4.1 lists the major smartphones with their WiFi chip sets and output power at the chip. We can see that most smartphones use the same chip set and that their average output power is between 16 dBm to 17 dBm for 802.11b. This is 3 dBm below the limit of 20 dBm (100 mW) defined by the standard. The output power for 802.11g and 802.11n are lower than 802.11b and between 10 dBm and 15 dBm.

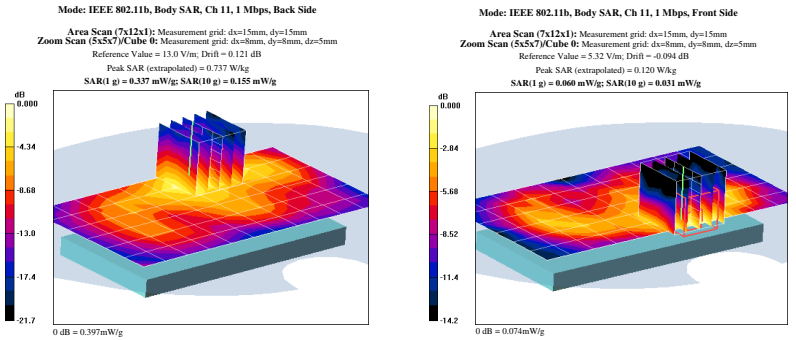
Phone Model	WiFi Chip set	Average RF Output Power (dBm)
HTC Nexus One	BCM4329	802.11b: 16.85-17.40 dBm
Samsung Galaxy Nexus	BCM4330	802.11b: 15.5-16.5 dBm
Samsung Galaxy SII	BCM4330	802.11b: 17.15 dBm
Samsung Galaxy SIII	BCM4330	802.11b: 16.0-17.12 dBm
Apple iPhone 4S	BCM4330	802.11b: 17.05 dBm

Table 4.1: *Smartphone WiFi characteristics (source: FCC test reports available at* `transition.fcc.gov/oet/ea/fccid`*).*

The antenna gain will impact the emitted and received power. With PiFa antennas the radiation is away from the ground plane (towards the rear of the phone) and the energy is directed away from the head. This also means that the maximum gain or energy (or best configuration for phones to communicate) is when phones' rear sides face one another.

The field intensity measurements performed by the FCC provides a visual representation of the antenna radiation pattern. Figure 4.3 shows the front and back measurements off the Nexus One phone.

Table 4.2 shows the surface area and the field intensity E measured by the FCC at the level of the antenna. The Nexus One emits 13 V/m, 5.32 V/m and 10 V/m at the rear, front and side, respectively. We can already see a difference of more than 7 V/m (more than double) between the front and rear. In order to translate the field intensity to the power density P_d, we compute



(a) Radiation pattern - Back

(b) Radiation pattern - Front

Figure 4.3: *EM Radiation pattern for the Nexus One (source: FCC).*

the power density from the field intensity by

$$P_d = \frac{E^2}{Z_0} = \frac{E^2}{120\pi} = \frac{E^2}{377} \, (Watts/m^2) \tag{4.3}$$

For $E = 13\ V/m$ (rear), we have $P_d = \frac{13^2}{377} = 0.45\ W/m^2$ which translate to $-3.5\ dBW/m^2$. For $E = 5.32\ V/m$ (front), we have $P_d = -11.2\, dBW/m^2$ and for the side $E = 10\, V/m$, we have $P_d = -5.76\, dBW/m^2$. This is a difference of almost 8 dB in power density between the rear and the front of the phone and 1.6 dB between the rear and side. From Table 4.2, we see that the power density on the side of the phone is usually very close to the back since the antenna is on the side (except for the iPhone 4S). Between the rear and the front (screen), differences in signal strength between 4.8 dB to 10.6 dB can be observed.

Another observation is that despite all phones having almost the same WiFi chip set and RF output power, they have different effective radiated power from the antenna depending on the side (i.e. rear vs. back. vs. side). This is due to the different antennas used in the different phones, which have different gains and different connector losses.

The still unknown feeder and connector losses between the WiFi chip and the antenna (B_T and B_R) and the propagation loss L will be evaluated using our outdoor measurement results. Before we present those measurements in Section 4.5, we remind the reader about the propagation models we will

Phone Model	**Surface area** cm^2	**Back** V/m	**Front** V/m (ΔdB)	**Side** V/m (ΔdB)
Nexus One	71	13.0	5.32 (7.7)	10.8 (1.6)
Galaxy Nexus	92	8.44 (3.75)	4.60 (5.27)	7.3 (1.26)
Galaxy SII	71	8.97 (3.22)	4.82 (5.39)	7.84 (1.17)
Galaxy SIII	96	10.5 (1.85)	3.1 (10.6)	5.17 (6.15)
iPhone 4S	67	11.5 (1.06)	6.61 (4.8)	8.58 (2.54)

Table 4.2: *Phone surface area and field intensity (back vs. front vs. side). In blue, for a given phone, the loss on other sides compared to the back side. In red, differences of the different phones compared to the Nexus One for the back.*

compare our measurement to in Section 4.4.

4.4 Propagation models

Propagation models fall into two categories: large-scale and small-scale models. Large-scale propagation models predict the mean signal strength for a given transmitter-receiver separation distance. The most well-known models are the Friis and the Log-Distance path loss models. Small-scale propagation models characterize the rapid fluctuation of received signal strength over a short time duration due to multi-path propagation (wave reflected by obstacles such as the ground and walls) or to motion (Doppler effect). Here we will not consider any slow or fast fading nor interference. We will however consider the two-ray ground model, an extension of the Friis model, which accounts for a direct wave and a ground reflected wave.

The Friis free space path loss model is described by Equation 4.4:

$$L_{FS}(dB) = 10\log_{10}\left(\frac{\lambda}{4\pi d}\right)^2 \tag{4.4}$$

where:
λ: wavelength (m)
d: TX-RX distance (m).

In the two-ray ground model, the total received energy is modeled as the vector sum of the direct transmitted wave and one ground reflected wave as illustrated by Figure 4.4. The model used for the propagation of the direct and reflected wave is the Friis model. The two waves are added constructively or destructively depending on their phase difference at the receiver. The magnitude and phase of the direct transmitted wave varies with distance traveled. The magnitude of the reflected wave depends on total traveled distance and the reflection coefficient (Γ) relating the wave before and after reflection.

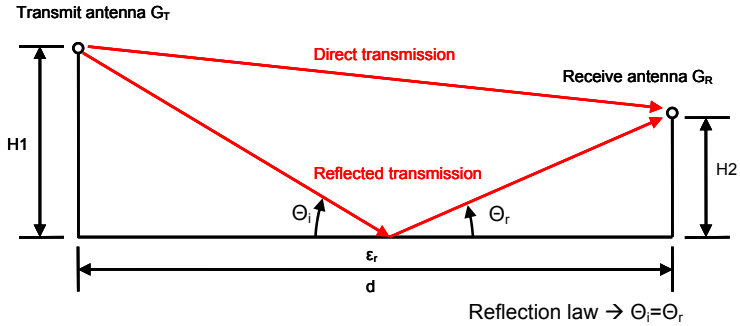


Figure 4.4: *Two-ray ground transmission model with LoS and ground reflection.*

Where the reflected wave hits the ground, only a portion of the energy is reflected and it mainly depends on the incident angle (Θ_i) and the the different dielectrics i.e., permittivity (E_r), permeability (μ_r) and conductivity (σ). Von Hippel [101] characterized a broad range of materials regarding their electrical properties. Based on the determined characteristics for typical ground, we will use $\varepsilon_r = 12$ for the relative permittivity (which includes E_r and μ_r) and $\sigma = 1$ for the ground reflection coefficient.

The Friis and two-ray ground models presented here will be used in the following Section 4.5 to obtain the remaining unknown parameters for the smartphone link budget.

4.5 A realistic smartphone propagation model

To obtain realistic values for the smartphone link budget elements, we use extensive outdoor measurements that are presented in this section. The results

of the different measurements are then used to derive the feeder and connector losses as well as a realistic path loss model for opportunistic networking in open (e.g. rural) spaces.

4.5.1 Experimental setup

The outdoor measurements took place in Dietikon (AG), Switzerland on a road with 450m of LoS (line of sight) surrounded by fields with low WiFi interference. A HTC Nexus One acting as an AP was mounted on a tripod at 1.25 m above ground with its rear side facing the LoS. An experimenter was holding different smartphones at 1.35 m facing the AP (i.e. rear facing the AP to have maximum antenna gain configuration). The experimenter walked backwards until out of range and then walking forward on the reverse path towards the AP. The experimenter phones acting as RX were receiving WiFi probes sent by the AP acting as a TX and measuring the RSS.

To get precise measurements, we took measurements every 0.5m from [0m,20m] using a tape measure and then every 10m from [20m,100m] and every 30 to 50m from [100m,350m] using a laser rangefinder ($+/-1m$ accuracy). Our measurements were limited by the range of our laser rangefinder to 355m although we still were within range of the AP beyond this distance as we will see in the next Section.

4.5.2 Propagation model fitting

The Figures 4.7 show the empirical measurements and the best fit with the Two-Ray model. They also show the Friis model with the parameters corresponding to the Two-Ray fit.

Figure 4.7a shows the fit of the model for the short range behavior 0-50m where $d > h_{TX}+h_{RX}$. The Two-Ray model was fitted with the following parameters: $Er = 12, h_{Rx} = 1.25m, h_{TX} = 1.35m, G_{TX} = G_{RX} = 1.1dBi$. The green plain line is the best fit with $P_{TX} = 7mW$ (8.45 dBm). The dashed green line is the corresponding Friis model with the same parameter. We can clearly see the impact of the ground and the difference between the RX signal power predicted by the Friis model and the Two-Ray mode. At short ranges the two-ray fit is striking. The signal experiences deep fades of up to 10 to 15 dB.

We look next at the long range behavior where $d \gg h_{TX}+h_{RX}$ in Figure 4.7b. We can clearly see the impact of the reflected wave which drastically

Figure 4.5: *Measurement location in Dietikon countryside with open fields and flat area. Position A is where the static AP is placed while B indicates 400m*

limits the reception range. The horizontal line at -97 dBm indicates the RX sensitivity threshold. This means that in ideal LoS free space conditions, one could detect beacons up to 1200 or 1500 m.[1] Due to the reflected wave on the ground (in phase opposition) for a horizontal LoS, this range reduces to 380-450m as we will see next. This is a radio range reduced by approximately 3 times compared to free space propagation.

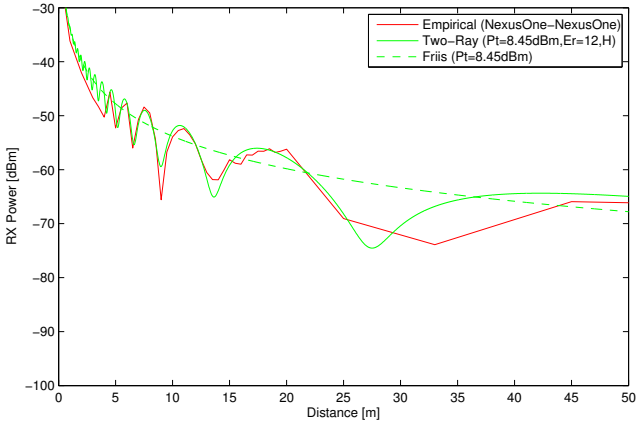
The Two-Ray model does not however fit the empirical data for the long-range part from 175m to 350m where one can observe a bump. This is due to the road going slightly up from 200 to 300m, which increases the reflection angle and the height of the carried phone. With the Two-Ray model, increasing the height of either TX or RX drastically increases the range and allows coming closer to free space propagation (Friis model). In the case at hand, we are even above what Friis predicts. This is due to the height leading to a con-

[1]This was confirmed by a joint Uepaa/Rega experiment with a phone on the ground and another hooked below a helicopter going up vertically.

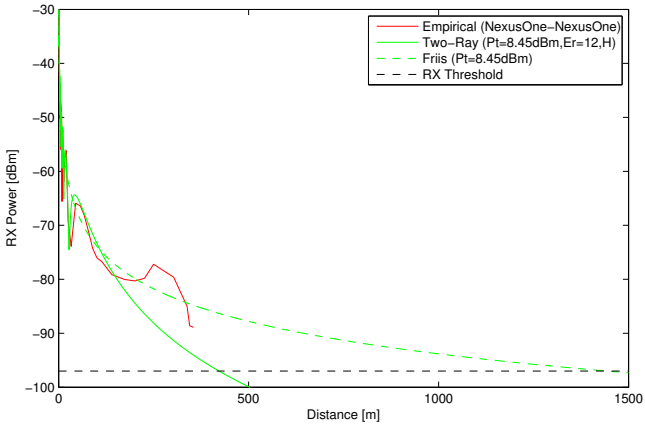
(a) Measurement setup

(b) Measurement line of sight

Figure 4.6: *Measurement setup, procedure and line of sight (>400m).*



(a) LoS Measurements (0-50m).



(b) LoS Measurements (0-1500m).

Figure 4.7: *Line of sight measurements with two HTC Nexus Ones fitted with the Two-Ray and Friis model.*

structive direct wave and reflected waves adding up at RX so that more power is received than what Friis would predict. Similar to the deep fades observed at short ranges due to destructive signals, this bump is due to constructive signals.

4.5.3 Feeder and connector losses

Fitting the two-ray ground model to our measurements outputs a P_T of 7 dBm as opposed to 17 dBm reported by the FCC (see Table 4.1). The difference of 10 dBm is due to the feeder and connector losses from the WiFi chip to the PiFa antenna. Based on our output power estimation by the model fitting, we conclude that the combined feeder and connector losses for sender and receiver are on the order of 10 dB (L_T+L_R). Thus the combined feeder and connector losses for the transmitter or receiver are 4-5 dB. The maximum range measurements reported in section 4.7 confirm those losses. For these range tests, we used a Samsung Galaxy Nexus as an AP and different phones as stations. The different phones could detect the AP beacons until different distances from the transmitter. Since TX was the same across all tests (i.e. same P_t and G_t) and the WiFi chip set of all RX phones are the same (i.e. BCM4330 - see Table 4.1), the observed range difference can only be explained by different antenna gain G_R and connector/feeder losses L_R.

4.6 Body attenuation

In this section, we investigate the impact of the smartphone carrier's body on the link budget.

4.6.1 Body attenuation model and measurements

During our early field measurements, we noticed the impact of the body attenuation on the received signal strength. The attenuation of the body blocking the LoS is often neglected when considering signal propagation involving mobile devices. When computing the attenuation of human tissue, we see that the attenuation is significant, as we will show in the following. To start off, we need to take the electrical properties of human tissue into account, as this defines how the wave will be influenced when it hits and traverses the body. From RF dosimetry measurements at 2.4 GHz based on human tissue equivalent material in [55], we use the dielectric constants reproduced in Table 4.3.

α and β are the real and complex components of the propagation constant γ

Table 4.3: *Dielectric properties of human tissue according to [55].*

	Relative permittivity	**Conductivity**	α	β
Torso	39.2	1.8	53.5	320.7
Head	52.7	1.95	50.2	369.5

($\gamma = \alpha + j\beta$). Those values will be used in Equations 4.5 and 4.6.

In order to assess the impact of the body on the LoS propagation, we need to consider three effects. First, the incoming wave will be partially reflected by the body. Second, the remaining part will be attenuated due to the velocity change over the air/body interface. Third, the wave will be attenuated depending on the distance traveled through the body. A part of the incoming wave is reflected by the body. The remaining part of the wave experiences loss due to velocity change at the air/body interface according to Equation 4.5.

$$L_\alpha = 20\ \log_{10}\left(\frac{\lambda_0}{\lambda}\right) = 20\ log_{10}\left(\frac{\frac{c}{f}}{\frac{2\pi}{\beta}}\right) = 20\ log_{10}\left(\frac{c}{f}\frac{\beta}{2\pi}\right) \tag{4.5}$$

where: c: speed of light.
f: frequency of the signal.
β: complex component of the propagation constant.

Once the wave traverses the body, Equation 4.6 defines the propagation losses through the body tissue where d_{body} is the distance the signal travels though the body..

$$L_\beta = 20\ \log_{10}\left(e^{\alpha d_{body}}\right) = 20\ \frac{\ln(e^{\alpha d_{body}})}{\ln(10)} = \frac{20}{\ln(10)}\alpha d_{body} \tag{4.6}$$

where: α: real component of the propagation constant.

A numerical evaluation of the resulting attenuation due to the propagation through the body gives 5 dB/cm of attenuation. Considering the highest allowed WiFi output power level at 20 dBm and a typical receiver sensitivity

between -90 and -95 dBm leaves a margin of approximately 110 dB for body attenuation, without taking any other propagation effects into account. The 110 dB body attenuation margin is already reached after traversing approximately 22 cm of body tissue. In practice, path loss and reflection by the body tissue further attenuate the waves, which causes the wave to be attenuated below reception threshold after traveling through the body for an even shorter distance. Therefore, we conclude that communication from the front to the back of the body through the body itself is not possible.

Diffraction

As transmission through the body is ruled out, another additional propagation effect comes into play: diffraction around the body. The rays traveling around the body due to diffraction suffer a noticeably smaller attenuation. The attenuation of waves which are diffracted around the body has mainly been studied in the context of body area networks, where antennas are placed on or near the body. [33] reports an attenuation in the order of 2 dB/cm at 2.4 GHz.

For our model, the body is assumed to be a perfect cylinder with a radius of 15 cm and with the dielectric properties of Table 4.3. It is illustrated in Figure 4.8. The attenuation of the body with a source away from the body and a receiver on or close to the body is then computed according to two rays being diffracted around the body, one around the top and one around the bottom of the body as shown in Figure 4.8 with the two dashed lines. We hence consider the values in [33] to estimate the attenuation due to diffraction according to Figure 4.8.

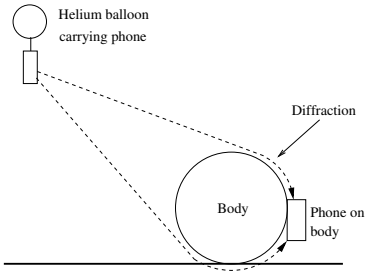


Figure 4.8: *Diffraction body model*

Diffraction measurements

We carry out measurements to validate this model. The setup is illustrated in Figure 4.9. We used two helium balloons to position the transmitting phone in the air and measure the attenuation of the body with a receiving phone placed on the experimenter's body. The experimenter is lying on the ground, so that the ground reflectionis negligible and only the diffraction can be measured.

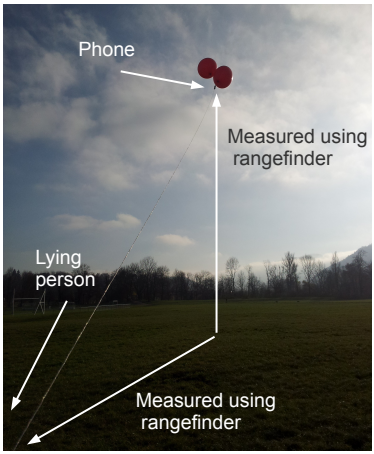


Figure 4.9: *Diffraction measurement setup*

The balloons were moved away from the body at a height of 15m to 20m. For each measurement point the distance from the person lying on the ground and the height of the balloon was measured.

As can be observed in Figure 4.10, the variations of the measurement are very large. Due to the wind, the position of the balloon could not be controlled perfectly. Therefore the measured distances do not exactly correspond to the actual distances during the measurements. We hence accounted in the model for a variation in transmitter position of 1m in each direction. Since the body is not an actual cylinder, the placement on the model cylinder had to be estimated, therefore a variation on the position of the phone on the model cylinder of 15° was also taken into account for the model. This shows that the body attenuation model is also strongly variable with small changes of position. We can see that the model generally overestimates the measurements.

This is due to the neglected orientation and polarization mismatch because

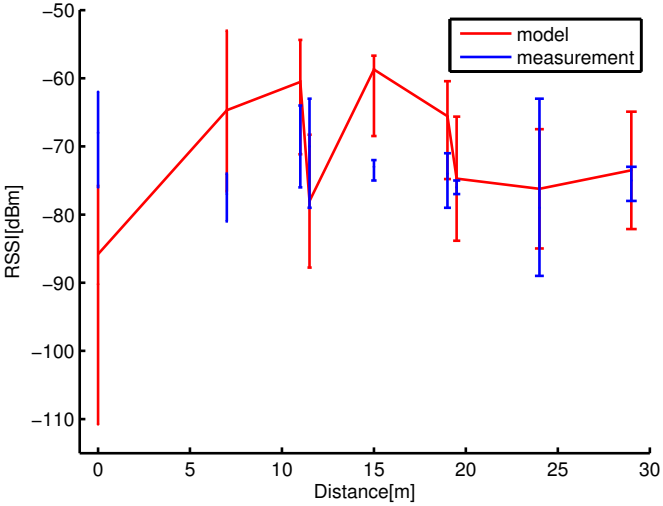


Figure 4.10: *Body diffraction measurement results*

of the relative position of the phones. Therefore we use an additional 10 dB of attenuation in the model to account for the orientation losses due to the phone being carried on the body in order to eventually match the measurements. The bottom line is that we consider our measurements to confirm the 2 dB/cm diffraction losses at 2.4 GHz.

4.6.2 LoS measurements with and without body attenuation

These LoS measurements were performed at the same location as for the evaluation of the propagation model. The measurements were however not taken at predefined intervals as before, but while slowly and continuously walking away from the access point phone. Additionally to checking the distance with the tape measure and the laser rangefinder, we were recording GPS data on the phone. This is why we get more fine-grained empirical plots. Figure 4.11 shows the received signal strength with and without the body attenuation as well as the difference between the two at the top. The plot without the body had the experimenter walking away from the AP with the phone's rear held

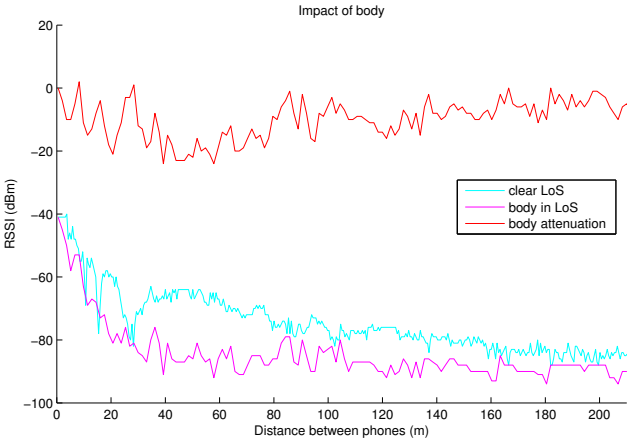


Figure 4.11: *LoS measurements w/ and w/o body.*

towards the AP and no body between the AP and the phone. The plot with body in-between had the experimenter walking away from the AP, holding the phone in the same position as before, but with the experimenter's body between the AP and the phone. A first observation is that the body attenuates the signal by a maximum of 20 dB and that the resulting measurements follow almost the same trend as the measurements without the body. The clear line-of-sight curve shows deep fades at short ranges as expected from the two-ray ground model. The curve with the body blocking the line-of-sight does not show the same explicit deep fades. This is caused by the interaction of the body with the incident waves. The direct wave is reflected to a larger extent than the ground reflected wave, due to the different incident angles on the body surface. The direct wave hits the body almost perpendicularly and most of the energy is reflected and less diffracted while the reflected wave has a lower angle which results in more energy being diffracted around the body. Regarding long ranges, we can observe that the difference between the two measurements tends to zero with distance. This is due to the fact that the further away from the AP, the less distance both waves (direct and reflected) have to travel around the body. At 100m we observe almost a 20 dB difference, which suggests a distance of 10cm (at our estimated loss of 2 dB/cm, see Section 4.6.1) traveled by the diffracted wave around the body. At 150m,

we only have 10 dB difference suggesting that the distance traveled around the body is half i.e., 5cm.

4.7 Maximum range and link capacity

Now that we have characterized all smartphone link budget elements, we will try to answer the following question: "How much good put can be achieved in a typical opportunistic networking scenario?". To be precise, we will estimate the good put of the contact when two pedestrians are walking past each other in opposite directions. Following the same methodology, we carried further measurements with different phones carried by the moving experimenter to assess the maximum range between two smartphones and the achievable WiFi transmission rates in this scenario. We will use those results to estimate the achievable good put for different walking speeds. According to Figure 4.7b, the Two-ray model predicts a maximum range of 415m for the HTC Nexus One. As a visual comparison, the Eiffel tower's last floor and the Empire state building altitudes are 279m and 381m, respectively.

4.7.1 Maximum range

We first walked backwards (facing the AP) until we lost connectivity to the AP and then walked back (forward) towards the AP. Table 4.4 gives the range at which the different phones were still able to receive WiFi beacons.

Table 4.4: *Beacon reception range.*

Phone	Out of range (backwards)	Into range (forward)
Galaxy Nexus	420m	358m
Galaxy S II	300m	270m
Galaxy S III	435m	405m

We can observe two things:

- Range Heterogeneity over different phone models,
- Asymmetry of In- and Out-range Border for the same phone.

Range Heterogeneity over different phone models: The first observation is the heterogeneity between the different devices. This is clearly due to the difference in antenna characteristics since all three devices in our measurements

embed the same WiFi chip set with similar performances (see Table 4.1). An additional important factor is the size of the phone and the antenna placement. This is highlighted by the Galaxy SIII which exhibits the largest range despite average values for its EM field on the different phone sides. Its size allows to have a larger ground plane (i.e. entire phone comprising the board and touchscreen) for the PiFa antenna coming close to the half-wavelength (6.1 cm) or wavelength (12.2 cm) of WiFi 2.4 GHz required to radiate EM with maximal gain. Besides the WiFi PiFa antenna placement (top right), was not obstructed by the experimenter fingers or palm. Also given its larger surface, less surface was covered with the experimenters hand and hence more power was received.

Asymmetry of In- and Out-range Border for the same phone: We can observe the asymmetric behavior of walking backward vs. forwards i.e. the range up to which we can communicate is much larger than the range at which devices can start to communicate when initially out of range. This range difference can be explained by the fact that when out of range, the device has to discover the AP by active probing (scanning with probe requests). The default period at which devices look for available APs is too large for our opportunistic networking based scenario.

Besides, as explained earlier, different phone characteristics might create asymmetric links i.e. one device can successfully send packets in one direction but the other way around might not be true. In the case at hand, our mobile devices could receive UDP beacons to a large range using the same PHY rate/modulation as AP MAC beacons so AP beacons should have been received successfully even before we could reconnect in practice. We can hence only explain the difference between the out of range border and within range border by the default scanning period (or passive listening period).

We hence looked into the scanning behavior to reproduce the outdoor measurements and understand the observed asymmetry. By reproducing the experiment and looking at the behavior of a Nexus One, a SII and a SIII, we found that phones only scan from time to time when they lost connection to the current AP (table Table 4.5).

4.7.2 802.11 PHY data rates

In addition to the maximum range, we need information about the rates (modulation and coding schemes - MCS) at which data was sent depending on the distance and SNR. Using the same measurement procedure as described

Phone	Scanning intervals w/ Screen On	Scanning intervals w/ Screen Off
Galaxy Nexus (2.3.6)	Scanning every 15s	10s, 300s, stop
Galaxy S II (4.0.3)	10, 10, 15, 10, 20, 32, 60, ...	30, 30, 30, 30, 10, 55, stop
Galaxy S III (4.0.4)	10, 10, 15, 32, 65, 12, 25, 10, ...	10, 10, 15, 92, stop

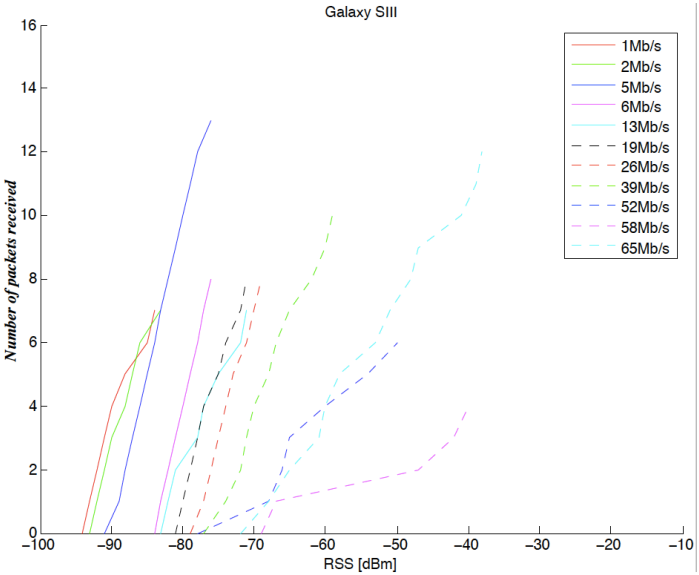
Table 4.5: *Smartphone scanning behavior.*

above, we plot the 802.11 PHY rates vs. RSSI for a Samsung Galaxy S III phone as shown in Figure 4.12. We used a Galaxy S III as it implements 802.11n and is currently one of the most widely spread Android smartphones. We also plot the 802.11 PHY rates as implemented in NS-3 vs. RSSI (and additionally the number of packets successfully decoded, i.e. with no bit errors, per RSSI and data rate). Note that the NS-3 simulator was calibrated with chip/bit error rate formulas relative to the different modulation as reported in [74].

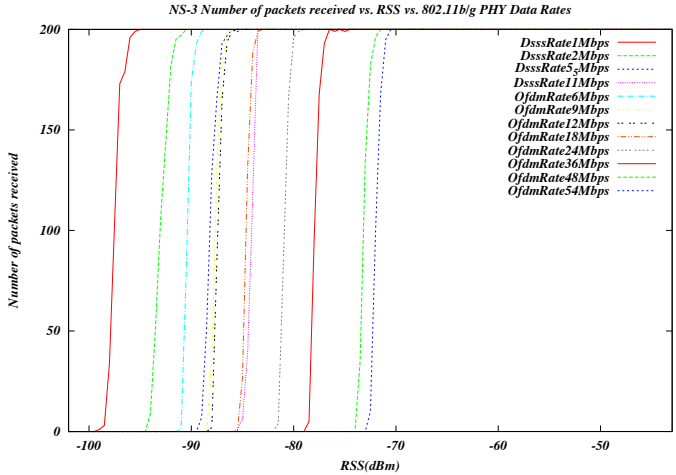
The overall behavior is as expected i.e. data rates correlate with the RSSI and the better the RSSI, the higher the data rate. The empirical data rates show similar behaviors across the different phones. The main difference is with the Galaxy SII, although embedding the same chip as the others, which for a given data rate operates at a higher RSSI compared to the Galaxy Nexus and Galaxy SIII. This explains also why the SII lost connectivity earlier compared to the others when walking out of range of the AP. The theoretical models of the data rates as implemented in NS-3 are ordered according to the RSSI. The empirical behavior as shown by the WiFi chip sets show less consistent and predictable behaviors as the MCS sequence order is not always respected.

4.7.3 Opportunistic WiFi link capacity of crossing pedestrians

Based on our evaluation, we are finally interested in the data transfer capacity while two pedestrians are crossing. In order to reflect the currently typical WiFi 802.11n standard implemented on smartphones, we base our capacity estimation on the data rates that we measured with a Samsung Galaxy S III phone. Measurements were taken using UDP unicasts (size 256 bytes) sent from the moving phone to the fixed access point phone. The data rates of



(a) Galaxy SIII (256 byte packets)



(b) NS-3 (256 byte packets)

Figure 4.12: *802.11 PHY data rates vs. RSSI for Galaxy S III and NS-3.*

received packets were recorded at the access point phone. This was done for approaching the access point (without body in between) and for walking away from the access point (with two user bodies in LoS) and thus one data rate series was obtained for each part of the crossing. To model the impact of different walking speeds, we calculate the relative fraction of time spent at each data rate for each part of the pedestrians crossing. The incoming transmission range is based on our measurements set to 350m and the range after the crossing is set to 80m. With the communication ranges obtained by our measurements we can estimate the capacity for different walking speeds. We assume that a variation of walking speed of +/- 0.2 m/s does not impact the 802.11 rate adaption. For walking speeds of 0.8 m/s (2.88 km/h or 1.79 mph), 1 m/s (3.6km/h or 2.24 mph) and 1.2 m/s (4.32 or 2.68mph) we get an estimated good put of 214, 171 and 143 MB respectively, taking the WiFi overhead and the packet errors we measured into account.

4.8 Related work

In this section we complement the information about related work that is already given directly in the previous sections. In [5] the authors address the directionality of smartphone antennas by proposing a multi antenna system that selects the best available antenna for the current transmission. Directional antennas in simulations was investigated in [6]. There are many commonly used propagation models that can also be applied to WiFi signals e.g. [46]. A survey of propagation models is given in [76]. [40] specifically use a propagation model based on Fresnel zones that takes the frequency and antenna height of smartphones into account. However, it neglects the fluctuations that we observed in our measurements which can be modeled by the two-ray ground model. Smartphone-based WiFi behavior is of interest, as in [82] a distance throughput model, that takes connection set up times into account is presented, however the body attenuation is not considered. The authors also show, that the DHCP discover process takes a significant time, which confirms the observation in our own measurements, that the communication range while walking towards an AP is lower than while walking away.

4.9 Discussion and conclusion

In this chapter, we investigated the feasibility of opportunistic communications using modern smartphones. We conducted a methodological study to evaluate WiFi performance of smartphones. First, we systematically characterized the smartphone link budget elements (antenna, WiFi chip set). We then focused on the case of two smartphones in outdoor LoS carried by pedestrians crossing in a low WiFi interference environment. We have found that body attenuation, usually overlooked in classical link budget formulation, can have an important impact since smartphones since are carried close to the body or held in the hand. Furthermore, we have demonstrated empirically and validated through the Two-ray propagation model that smartphones can communicate from 300m to 450m. This results in a good put of 143 MB for crossing of two pedestrians walking at 4.3 km/h. Eventually, an important finding is that the WiFi range is limited by the destructive ground reflective wave and that ideal LoS propagation could achieve more than 1km with current smartphones.

We did not further investigate the impact of the phone's relative orientation and the related antenna's horizontal vs. vertical polarization. Imagine two phones stacked on top of each other in the same orientation. In this case, the sender and receiver antennas are perfectly aligned regarding the polarization of the emitted and received wave. In general, the phones may be in an arbitrary angle towards each other, which may cause additional loss, as the emitted (polarized) wave is not aligned perfectly to the receiving phone's antenna. The mismatch prevents the receiving antenna from recovering all of the energy of the received signal. This loss is called polarization mismatch. Early measurements in a anechoic chamber did not provide consistent results. However, a worst case was observed with a maximum of 5 dB loss due to polarization mismatch. We did not investigate further the possibility for asymmetric links to happen due to the heterogeneous characteristic and performance of different smartphones.

Nonetheless, we believe our study already proves the feasibility of opportunistic networking in rural areas based on current smartphones' capabilities. Forthcoming standards such as LTE-Direct or 802.11ah will further extend smartphones' capabilities for such scenarios. LTE-Direct is a P2P extension of 4G, which will enable a very low power peer discovery up to 500m in a first release and data transfers in a second release. 802.11ah is an evolution of 802.11ac with low throughput (100 kb/s) utilizing the 900MHz band with

narrow bandwidth (from 1 to 16 MHz). The low frequency of 802.11ah aimed at rural areas will allow ranges up to 1km and more and the lower frequency band of 900 MHz will reduce the diffraction losses due to the carrier's body.

Most opportunistic contacts in rural areas will happen in social places such as villages or markets with a mid- to high-density of smartphones. There, interference will play a higher role on impacting the capacity of opportunistic networking. For future work we hence plan to evaluate high-density scenarios through real-world experiments and simulations. We plan on integrating all the findings of this work to improve the NS-3 simulator. Also, we plan to extend the Two-ray ground model to account for uneven or inclining terrain and more diverse environments. For now, we only considered a LoS environment i.e., path loss exponent of 2, and we plan to evaluate the path loss exponent of different environments such as forest or villages. Eventually, NS-3 needs to be fine-tuned to reproduce the rate adaption behavior of smartphones' WiFi chip set (here mostly BCM4330) as it is currently parametrized for the Prism 2.5 chip, which is quite outdated. We need to carry more experiments to better understand the switching behaviors between the different possible rates and further validate our capacity estimations empirically.

Copyright

Chapter 5

Conclusions

The research challenges around opportunistic networking are manifold. With this work, we believe to have contributed to better understanding of two areas of those challenges. First, we believe that the social aspect of opportunistic networking can be an important driver behind increased performance as well as commercialization. However, privacy is paramount as even on closed platforms like Facebook today, privacy violations lead to unintended and sometimes damaging consequences for users. A crashed party with 1000 instead of the originally intended 20 users is only one of the more harmless examples. Second, the reality of smartphones and how they are used is an important aspect as opportunistic networking needs to blend in seamlessly with the typical use of smartphones today.

Opportunistic networks should support the best available methodologies for routing information. Currently, informing the forwarding decision in opportunistic networks with the analysis of a contact graph that encodes social links is the most promising routing approach available. We have shown that the performance of this routing approach can be maintained with significantly less accurate information about the social links. Two observations led us to our approach to change the contact graph. On the one hand, opportunistic networks do not use end-to-end routing, but decide on each contact whether to forward data or not based on a simple comparison of a utility value that is calculated on the currently available graph. Second, with a properly designed algorithm, the graph can be obfuscated to a large extent without interfering with the utility calculation used in the forwarding decision. Based on those two observations, we designed an algorithm that modifies a given contact graph while maintaining the ranking of the nodes regarding a given routing metric. This effectively removes correct and superfluous information about social links. The algorithm even works with only limited graph knowledge, as nodes in an opportunistic networks would in reality be able to collect. When working with global graph knowledge, the graph size can become too large for our stepwise optimal algorithm. Thus, we designed a heuristic that allows the algorithm to process much larger graphs without a significant performance impact.

Another observation in opportunistic networks is, that many aspects like security, trust and rating can benefit from the inherent social structure and sometimes even more from explicitly created trusted (social) links. This is where privacy protection and utility of social information work against each other. Thus, new mechanisms are required to have privacy protection and the explicit use of social information coexist in one environment. With our pri-

vacy preserving social link detection protocol, we introduced a practical way of keeping social links private without sacrificing the scarce resource (contacts) in opportunistic networks. In an anonymous environment, two nodes can determine whether or not they share a pre-established social connection with each other. An attacker can neither learn anything about the social links of a single node, nor relate two friendship encounters with each other. We achieve this strong probabilistic protection by using two Bloom filters with different hash functions and randomly distributing the set of social link identifiers of each node across them. Additionally, the social link identifiers are mutated over time and noise is added to the Bloom filter, such that an attacker can fingerprint a user with negligible probability. Furthermore, the computational and communication overhead are kept at a minimum with typically 2 - 4 messages and a few kilobytes of data.

Finally, contacts are the base for any interaction in opportunistic networks. While mobility is a major and well investigated factor for contacts, implementations of opportunistic networks have to deal with an additional component. Smartphones are the predominant device for opportunistic networking. How smartphones are built and used does have a significant impact on the radio contact. For example there is a significant difference in signal level and thus data rate and communication range if two smartphone users are walking towards or away from each other. We investigate the link budget and body impact by collecting publicly available information about smartphones, modeling and measurements. By combining those sources of information, we are able to characterize all elements that are involved in the radio communication between two smartphones. This covers everything from the transmitter, connection to and the antenna itself, a widely used propagation model and also a simple body model to model the impact of the users body on the propagation.

5.1 Critical assessment

5.1.1 Privacy for contact graph based routing

With social privacy and protecting social links in contact graph based routing, one has to put things into a larger perspective. The general approach that we use in this part of work is graph transformation with feature preservation. In other words, we aim at changing as many graph properties as possible while keeping a specific graph feature present and accurate in the changed

graph. This is the price to pay for and the limit of supporting contact graph based routing. Attacks that target the very nature of the routing utility calculation can not be prevented in such a setting. A typical attack that can not be prevented is to identify a given number of top ranking nodes regarding the routing metric. The success of such an attack will degrade proportionally to the correctness of the routing decisions. Nevertheless, our algorithm succeeds at hiding the real social links from the attacker and thus achieve its goal of removing available but not explicitly required information from the graph. Due to the sparse nature of contact graphs, it is also not trivial for an attacker to invert the algorithm and thus reconstructing the original graph. For each of the few real edges, there is a large selection of candidate edges to insert, even if the ranking of the nodes has to be maintained. The algorithm can thus for each edge choose from a large set of destination edges. Additionally to maintaining the ranking of the nodes, the attacker has to pick the correct edge from the set of edges that would result in the same ranking.

5.1.2 Privacy preserving social link detection

Detecting social links in an anonymous environment opens up a large field of potential benefits for opportunistic networking services. The contacts themselves though are a scarce resource and as much contact time as possible should be available for content transfer. Detecting friends takes away some of the contact time. While a single detection is very efficient, there is another drawback caused by the anonymous setting itself. If multiple contacts are available, with the given proposal, a connection to each node must be established and the protocol must be run to detect an existing social link. If a node has many contact opportunities and a connection to a social link is preferred for some reason, the protocol has to be run with one node after another (or in parallel) until a social link is identified. In this case, the system is vulnerable to a Sybill attack, where an attacker creates a large number of virtual neighbors and thus decreases the chance of a legitimate node to connect to another legitimate node. Furthermore, the Sybill nodes can have modified Bloom filters (all bits set in the worst case) that cause a high number of false positives. This in turn forces the legitimate nodes to move to the authentication phase for each Sybil node. A small change in the protocol could improve the situation. If nodes broadcast their Bloom filter whenever they detect a new neighbor, the legitimate nodes can save some time and pre test the available Bloom filters and only connect to potential social link nodes. Also, since the average number of bits set in the Bloom filter is about constant for the

system, malicious Bloom filters can simply be ignored. In a fully anonymous environment, a sybil attack [22] (as sophisticated denial of service) can not be fully mitigated. Finally, there is the possibility that advances in cryptography, crypto protocols and smartphone computational capabilities reach a point where solutions based on privacy preserving matching become feasible to be run on smartphones. A solution that provides cryptographic guarantees to the privacy protection at the same or similar performance of our probabilistic protection is clearly preferable. Until that day comes, our protocol will enable privacy preserving friendship detection for implementation in current generation smartphone based opportunistic networking.

5.1.3 Smartphone WiFi contact properties

The study of contacts and their properties in opportunistic networking is traditionally focused on effects caused by mobility of the users. Since antenna characteristics and propagation models are well known, one could ask what insight the investigation of WiFi based smartphone to smartphone communication could bring. Some feedback we received left the impression, that there is no room for research in this direction. Indeed, it is tempting to assume that there is nothing to investigate, since everybody has a smartphone and WiFi just works fine in most peoples experience. Thus, smartphones would be just as well suited for opportunistic networking than for their traditional purpose. While it is true that there is a fair amount of research about antenna design and propagation models available, there are still many unknowns in the transmission chain. This is exactly what our work details out. The antenna placement and directionality are subject to many optimizations, such as available space on the phone, directed transmission away from the expected position of the body (or head) of the user in order to minimize radiation exposure of human tissue and at the same time optimize signal quality. Another argument is the exemplary use of a few models of smartphones that are currently available. The characteristics could change rapidly from one smartphone generation to the next and our results would be worthless. While this could be true for future smartphone generations, it turns out that the last few generations of smartphones not only share a similar basic layout regarding antenna placement, also the chip sets used are very few in number. This might be due to design criteria such as directing radiation away from the head as well as market forces (chip set manufacturers). Since we do not expect the general smartphone design criteria to change for future smartphone generations, our work will remain useful to understand contact properties. A simple exam-

ple being a significant range difference from a few hundred meters when two pedestrians are walking towards each other (with smartphones facing each other back to back) to a few ten meters after having crossed and walking away from each other (two bodies in between and smartphones facing screen to screen).

5.2 Future work and outlook

5.2.1 Opportunistic network privacy

Due to the user device correspondence and the fact that mobile phones are almost always on the user, privacy in opportunistic networks is a complex topic. The social privacy aspect that this work contributes to is the one aspect that has received less attention than the other aspects so far. The social information that is available in opportunistic networks is both a blessing and a curse. Some say "ignorance is bliss" but for opportunistic networking technologies, this is neither true for the social information itself, neither for privacy.

Many privacy issues have already been addressed and while a few privacy issues remain we believe that in order to further the development and implementation of opportunistic networks we need to take a step back and another look at the existing privacy protection proposals. While it is tempting to address individual privacy problems in opportunistic networks, it seems much more challenging to design a solution that reliably covers all privacy needs. This requires to combine multiple of the existing solutions into one privacy protection framework for general purpose opportunistic networking. However, the interdependence, overlap and mutual influence of combining multiple privacy protection solutions are largely unknown today. Questions such as "how do a location privacy and social privacy solution interact with each other?" need to be addressed. As this thesis filled a gap of privacy protection for social information in opportunistic networks, such questions may now be addressed. It is easy to imagine a situation where a location privacy solution prevents a contact graph from being built successfully, because multiple contacts can not be attributed to the same user. There are generally three options how various privacy protection mechanisms can impact each other. When two mechanisms work truly independently, then there might be no impact at all. Otherwise, there is either a negative or a positive interaction. In the case of a negative impact, one mechanism adversely effects another, reducing overall privacy protection effectiveness. A positive interaction results in a sit-

uation, where the mechanisms improve each others performance. The major challenge for opportunistic network privacy protection will be how to protect the maximum number of privacy aspects with a minimum number of protection mechanisms. And this without adverse effects and while still supporting social links and interaction and all other envisioned services.

5.2.2 Where do opportunistic networks go?

Opportunistic networks are now a research topic for more than 10 years. The expectations in the beginning were high, yet there is no killer application for opportunistic networks. The focus now lies on how and where opportunistic networks can be widely and successfully deployed. A real deployment also has the potential to raise new research questions in the area. The development of new technologies starts to make some of the envisioned scenarios for opportunistic networks obsolete. One example is Internet for remote areas where Google and Facebook are venturing with balloons or drones to provide cellular connections to uncovered areas. The good news about this development is, that large companies do see a business case for such a scenario. The bad news is that opportunistic networking is not their technology of choice. The business model for the large companies depends on users accessing the Internet via their services. Opportunistic networks do not offer the same amount of control as centralized approaches to implement such a business model.

Circumventing censorship is a promising field where opportunistic networks can excel. Censorship in a fully decentralized environment is a much harder problem than with a central infrastructure. A recent example are the protest in Hong-Kong where the protesters used an application that opportunistically exchanged messages called FireChat. As those messages did not traverse the government controlled Internet, the protesters were able to exchange messages without being subjected to censorship. While we believe that opportunistic networks can provide a very strong solution, driving the technology on this track presents a severe risk. When a technology becomes predominantly known for avoiding censorship or other illegal uses, it risks to be outlawed. First, another use of opportunistic networks has to drive their popularity and circumventing censorship can sail through behind. Additionally, there is no business model as of yet to make money by circumventing censorship.

If rural areas and censorship are ruled out as driving forces, mainly safety

and entertainment remain as candidate drivers. Safety applications, such as disaster communication are hard to build business models around. Justifying expenses for a just in case safety application is not always easy and once a disaster strikes, it is too late to distribute the application to the target population. Thus the currently most promising scenario is to integrate opportunistic networking in the entertainment sector. Localized multi player gaming is one obvious application, where opportunistic networks can provide connectivity. Another one is opportunistic technology as enabler for art projects. There are already art projects that work in a GPS based manner. Creating art projects where interaction of the present users is part of the artwork (and thus always in motion) can provide fascination results. Many of the applications that seem to work on a specific location proximity today (e.g. Bump) are actually using server based infrastructure to determine which users are close to each other centrally. Such approaches are sometimes fooled by gps spoofing, especially in gaming. With opportunistic networking technology backing local interactions, cheating becomes much more difficult.

While the feasible application scenarios for opportunistic networks were reduced over the last years, there are still enough potential and promising scenarios where opportunistic networks can be established successfully. Ideally, opportunistic networks are made popular in applications where they can soft-fail, that means without causing significant financial or functional losses.

5.3 Uepaa - an attempt at commercializing opportunistic networking

During the course of this work, the author had the opportunity to participate in a collaboration with a startup company to build a smartphone application with opportunistic communication during a 14 month period. The project was sponsored by the Swiss government under the KTI/CTI[1] grants to transfer technology from universities to commercialization. The goal of the Uepaa application was to create an application for alpine safety. The Swiss mountains are only partially covered by cellular networks. However, the mountains are a popular place for free time and sports activities for a large group of the population. Especially hiking in summer and skiing and ski touring in winter are very popular. Unfortunately, there are also accidents that happen while

[1] Kommission für Technologie und Innovation `https://www.kti.admin.ch/kti/de/home.html`

people follow their activities. Getting help in a mountain area without cellular coverage can be difficult. Usually users do not know where to go to get into cellular coverage to call for help if they can move at all. This is where opportunistic communication, in this case based on WiFi-Opp [93] comes into play.

The Uepaa application uses opportunistic communication to detect neighbors as soon as the users leave the area with cellular coverage. When the opportunistic communication kicks in, data about the users and their gps locations are exchanged opportunistically among all Uepaa users in vicinity. As soon as any user regains a cellular data connection, the information about who the user met and where is transmitted to the Uepaa servers. A web interface allows to track registered users based on their own direct position reports as well as the opportunistically relayed data. In the case of an emergency, the opportunistic module is also used to broadcast an emergency beacon. As soon as another user receives the emergency signal, the application alarms the user and forwards the alarm to the Uepaa servers if an online connection is available, where the rescue operation can be initiated and coordinated.

Coming from the research community, there were many challenges to transfer the existing knowledge to the startup company. Many practical issues were raised and had to be solved. There were also questions where we did not find answers for in research literature and thus had to address them ourselves. In order to prove the feasibility of the application, we investigated how a smartphone WiFi based opportunistic network behaves in high density scenarios. Think of a hut in a skiing resort in the mountains where 400 people gather to eat and relax. This scenario easily overloads a single cell in a cellular network, but how does this affect opportunistic networking? We conducted a scalability study that takes the practical and technological features into account that are relevant in this case. This ranged from the theoretical WiFi capacity over all the restrictions of current Android and iOS versions to beaconing size and intervals. Practical limitations for WiFi-Opp on current Android and iOS are for example that the channel for the AP can not be selected on Android, that iOS does not allow to programatically create an AP or that the maximum number of clients for an Android AP ranges from 8-10 depending on the device.

Another task was to design a security architecture that prevents malicious users from injecting or modifying information in the system (especially the opportunistic part). Beyond what we presented in chapter 4 we also investigated the WiFi signal propagation in different snow conditions (if somebody

is buried under an avalanche) and from positions high up in the air, such as from a rescue helicopter that is searching for opportunistic emergency beacons to locate a victim more quickly.

Working at the interface between research and commercialization is not always an easy task. There is often a conflict of research methodology and time and financial constraints imposed by the business plan. From a business point of view, a fast and approximate answer is often preferred to a real understanding of the problem. On the other hand, the expectations are, that the research has covered all important aspects and no (negative) surprises are waiting down the road. Eventually, the Uepaa application was launched successfully and is available for iOS and Android in multiple countries across Europe.

Today, Uepaa is still maintaining the alpine safety application but is moving on towards providing an opportunistic communication platform that can be integrated with other applications and used by other programmers. The platform called p2pkit, draws from the experience gathered from the alpine safety application and extends the message and media types that can be exchanged opportunistically. All the improvements to the opportunistic communication that resulted from this work are now not only limited to alpine safety, but are available for a much wider set of use cases. The move away from application specific integration of opportunistic technologies towards a general purpose opportunistic framework underlines the potential expected of opportunistic communication in commercial settings. With Open Garden [69] there is already competition on the horizon which also aims at providing an API for opportunistic communication. Together with manufacturer pushed developments such as WiFi Aware [103] and LTE Direct [80] the potential for opportunistic networking on smartphones seems to gain momentum. Having an opportunistic SDK and API available opens up opportunistic communication to a large group of new users, that extends beyond classical app developers. Groups such as artists and others might think of new uses for opportunistic communication and drive its potential even further.

5.4 Publications

The course of this work led to the following publications:

- Distl, B., & Hossmann, T. (2014). Privacy in opportunistic network contact graphs. In Proceeding of IEEE International Symposium on

a World of Wireless, Mobile and Multimedia Networks 2014. IEEE. doi:10.1109/WoWMoM.2014.6919020

- Distl, B., & Neuhaus, S. (2015). Social power for privacy protected opportunistic networks. In Proceeding of Communication Systems & Networks (COMSNETS) 2015.

- Distl, B., & Legendre, F. (2015). Are smartphones suited for DTN Networking? In Proceeding of International Workshop on Wireless Network Measurements and Experimentation (WiNMeE) 2015

- Under submission: Distl, B., & Hossmann, T. (2015). Privacy protecting social links in contact graphs for opportunistic routing.

- Distl, B., & Csucs, G., & Trifunovic, S., & Legendre, F., & Anastasiades, C. (2010). Extending the reach of online social networks to opportunistic networks with PodNet. In Proceedings of the Second International Workshop on Mobile Opportunistic Networking (MobiOpp '10). DOI=10.1145/1755743.1755779

- Trifunovic, S., & Anastasiades, C., & Distl, B., & Legendre, F. (2010) PodNetSec - Secure Opportunistic Content Dissemination. ACM Mobisys, San Francisco, CA, USA, June 2010

- Trifunovic, S., Distl, B., Schatzmann, D., & Legendre, F. (2011, September). WiFi-Opp: ad-hoc-less opportunistic networking. In Proceedings of the 6th ACM workshop on Challenged networks (pp. 37-42). ACM. [The contribution of the author to this work was focused on energy measurements and evaluation]

Bibliography

[1] Bluedating. [Online, 2015-11-14] `http://en.wikipedia.org/wiki/Bluedating`.

[2] I. Aad, C. Castelluccia, and J.-P. Hubaux. Packet Coding for Strong Anonymity in Ad Hoc Networks. In *2006 Securecomm and Workshops*, pages 1–10. IEEE, Aug. 2006.

[3] M. M. Adam Kirsch. Less hashing, same performance: Building a better bloom filter. In *14th Annual European Symposium on Algorithms (ESA 2006)*, pages 456—-467, 2006.

[4] C. C. Aggarwal and H. Wang, editors. *Managing and Mining Graph Data*, volume 40 of *Advances in Database Systems*. Springer US, Boston, MA, 2010.

[5] A. Amiri Sani, L. Zhong, and A. Sabharwal. Directional antenna diversity for mobile devices. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking - MobiCom '10*, page 221, New York, New York, USA, Sept. 2010. ACM Press.

[6] E. Anderson, G. Yee, C. Phillips, D. Sicker, and D. Grunwald. The impact of directional antenna models on simulation accuracy. In *2009 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, pages 1–7. IEEE, June 2009.

[7] S. K. Belle and M. Waldvogel. Consistent Deniable Lying: Privacy in Mobile Social Networks. In *Workshop on Security and Privacy Issues in Mobile Phone Use (SPMU)*, Sidney, AUS, 2008.

[8] G. Bigwood, D. Rehunathan, M. Bateman, T. Henderson, and S. Bhatti. Exploiting Self-Reported Social Networks for Routing in Ubiquitous Computing Environments. In *2008 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, pages 484–489. IEEE, Oct. 2008.

[9] 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 J. Kay, P. Lukowicz, H. Tokuda, P. Olivier, and A. Krüger, editors, *Pervasive*

Computing, volume 7319 of *Lecture Notes in Computer Science*, pages 370–387. Springer Berlin Heidelberg, 2012.

[10] A. Bland. FireChat – the messaging app that's powering the Hong Kong protests. [Online, 2015-11-14] `http://www.theguardian.com/world/2014/sep/29/firechat-messaging-app-powering-hong-kong-protests`, sep 2014.

[11] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, July 1970.

[12] Bluetooth Special Interest Group. Bluetooth Specifications. [Online, 2015-11-14] `https://www.bluetooth.org/en-us/specification`.

[13] C. Boldrini, M. Conti, and A. Passarella. Exploiting users' social relations to forward data in opportunistic networks: The HiBOp solution. *Pervasive and Mobile Computing*, 4(5):633–657, Oct. 2008.

[14] S. Capkun, J.-P. Hubaux, and L. Buttyan. Mobility Helps Peer-to-Peer Security. *IEEE Transactions on Mobile Computing*, 5(1):43–51, 2006.

[15] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, E. Travis, and H. Weiss. Delay-Tolerant Network Architecture: The Evolving Interplanetary Internet (Draft). Technical report, 2002.

[16] A. Chaintreau, P. Hui, J. Crowcroft, and C. Diot. Pocket switched networks: Real-world mobility and its consequences for opportunistic forwarding. Technical report, University of Cambridge, 2005.

[17] K. Christoffer and S. Christina. Annual Incident Reports 2014. [Online, 2015-11-14] `https://www.enisa.europa.eu/activities/Resilience-and-CIIP/Incidents-reporting/annual-reports/annual-incident-reports-2014/at_download/fullReport`, 2014.

[18] Cisco. Cisco virtual networking index VNI Whitepaper. [Online, 2015-11-14] `http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html`.

[19] T. Clausen and J. Philippe. Optimized Link State Routing Protocol (OLSR). [Online, 2015-11-14] https://tools.ietf.org/html/rfc3626.

[20] E. M. Daly and M. Haahr. Social network analysis for routing in disconnected delay-tolerant MANETs. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing MobiHoc 07*, page 32. ACM, ACM Press, 2007.

[21] W. Dong, V. Dave, L. Qiu, and Y. Zhang. Secure friend discovery in mobile social networks. In *2011 Proceedings IEEE INFOCOM*, pages 1647–1655. IEEE, Apr. 2011.

[22] J. Douceur. The sybil attack. *Peer-to-peer Systems*, 2429:251–260, 2002.

[23] DTNRG. The IETF Delay tolerant networks (DTN) research group. [Online, 2015-11-14] http://www.dtnrg.org.

[24] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. Age matters. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing - MobiHoc '03*, page 257, New York, New York, USA, June 2003. ACM Press.

[25] N. Eagle and A. S. Pentland. {CRAWDAD} dataset mit/reality (v. 2005-07-01). Downloaded from http://crawdad.org/mit/reality/20050701, 2005.

[26] N. Eagle, A. S. Pentland, and D. Lazer. Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of Sciences of the United States of America*, 106(36):15274–8, sep 2009.

[27] ETH Zurich. Twimight - the mighty Twitter client. [Online, 2015-11-14] http://www.twimight.com/.

[28] ETSI TSGR. TS 136 300 - V12.7.0 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2 (3GPP TS 36.300 version 12.7.0 Release 12). 2015.

[29] EU FP7 N4C Project. Networking for Communications Challenged Communities. [Online, 2015-11-14] http://www.n4c.eu/.

[30] K. Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM '03*, page 27, New York, New York, USA, 2003. ACM Press.

[31] FCC. goTenna FCC ID 2ABVK02629. [Online, 2015-11-14] `https://fccid.io/2ABVK02629`.

[32] R. Fielding and J. Reschke. Hypertext Transfer Protocol (HTTP/1.1): Authentication. [Online, 2015-03-21] `https://tools.ietf.org/html/rfc7235`.

[33] A. Fort, F. Keshmiri, G. Crusats, C. Craeye, and C. Oestges. A Body Area Propagation Model Derived From Fundamental Principles: Analytical Analysis and Comparison With Measurements. *IEEE Transactions on Antennas and Propagation*, 58(2):503–514, Feb. 2010.

[34] M. J. Freedman, K. Nissim, and B. Pinkas. Efficient Private Matching and Set Intersection. In *Advances in Cryptology - EUROCRYPT*, volume 3027, pages 1–19. Springer-Verlag, 2004.

[35] L. C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215–239, Jan. 1978.

[36] K. Fukuda, M. Aoki, S. Abe, Y. Ji, M. Koibuchi, M. Nakamura, S. Yamada, and S. Urushidani. Impact of Tohoku earthquake on R&E network in Japan. In *Proceedings of the Special Workshop on Internet and Disasters - SWID '11*, pages 1–6, New York, New York, USA, dec 2011. ACM Press.

[37] Gartner. Gartner Says Smartphone Sales Surpassed One Billion Units in 2014. [Online, 2015-11-14] `http://www.gartner.com/newsroom/id/2996817`.

[38] S. Gibbs. Facebook 'tracks all visitors, breaching EU law' | Technology | The Guardian. [Online, 2015-11-14] `http://www.theguardian.com/technology/2015/mar/31/facebook-tracks-all-visitors-breaching-eu-law-report`, 2015.

[39] GoTenna. goTenna. [Online, 2015-11-14] `http://www.gotenna.com`.

[40] D. Green and A. Obaidat. An accurate line of sight propagation performance model for ad-hoc 802.11 wireless LAN (WLAN) devices. In *2002 IEEE International Conference on Communications. Conference Proceedings. ICC 2002 (Cat. No.02CH37333)*, volume 5, pages 3424–3428. IEEE, 2002.

[41] G. T. G. Greenwald. Revealed: how US and UK spy agencies defeat internet privacy and security | US news | The Guardian. [Online, 2015-11-14] `http://www.theguardian.com/world/2013/sep/05/nsa-gchq-encryption-codes-security`, 2013.

[42] S. L. Guillén, M. F., & Suárez. Explaining the global digital divide: Economic, political and sociological drivers of cross-national Internet use. *Social Forces*, 84(2):681–708, 2005.

[43] S. Guo, M. H. Falaki, E. A. Oliver, S. Ur Rahman, A. Seth, M. A. Zaharia, and S. Keshav. Very low-cost internet access using KioskNet. *ACM SIGCOMM Computer Communication Review*, 2007.

[44] B. Han, P. Hui, V. S. A. Kumar, M. V. Marathe, G. Pei, and A. Srinivasan. Cellular traffic offloading through opportunistic communications: a case study. In *CHANTS*, 2010.

[45] S. Hanhijärvi, G. C. Garriga, and K. Puolamäki. Randomization techniques for graphs. In *SIAM Conference on Data Mining (SDM)*, pages 780–791. SIAM, 2009.

[46] M. Hata and B. Davidson. A report on technology independent methodology for the modeling, simulation and empirical verification of wireless communications system performance in noise and interference limited systems operating on frequencies between 30 and 1500 mhz. In *TIA TR8 Working Group, IEEE Vehicular Technology Society Propagation Committee*, 1997.

[47] T. Hossmann and P. Carta. Twitter in disaster mode: security architecture. In *ACM CoNEXT SWID*, 2011.

[48] T. Hossmann, F. Legendre, and T. Spyropoulos. From contacts to graphs: pitfalls in using complex network analysis for DTN routing. In *INFOCOM Workshops*, 2009.

[49] T. Hossmann, G. Nomikos, T. Spyropoulos, and F. Legendre. Collection and analysis of multi-dimensional network data for opportunistic networking research. *Computer Communications*, 35(13):1613–1625, July 2012.

[50] T. Hossmann, T. Spyropoulos, and F. Legendre. Putting contacts into context. In *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing - MobiHoc '11*, page 1, New York, New York, USA, May 2011. ACM Press.

[51] C.-K. Hsieh, H. Falaki, N. Ramanathan, H. Tangmunarunkit, and D. Estrin. Performance evaluation of android IPC for continuous sensing applications. *ACM SIGMOBILE Mobile Computing and Communications Review*, 16(4):6, feb 2013.

[52] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. In *MobiHoc*, pages 241–250. ACM, ACM, 2008.

[53] E. Hyytiä, J. Virtamo, P. Lassila, J. Kangasharju, and J. Ott. When does content float? Characterizing availability of anchored information in opportunistic content sharing. *Proceedings - IEEE INFOCOM*, pages 3137–3145, 2011.

[54] H. Jun, M. H. Ammar, M. D. Corner, and E. W. Zegura. Hierarchical power management in disruption tolerant networks with traffic-aware optimization. In *Proceedings of the 2006 SIGCOMM workshop on Challenged networks - CHANTS '06*, pages 245–252, New York, New York, USA, sep 2006. ACM Press.

[55] M. Kanda, M. Ballen, S. Salins, C.-K. Chou, and Q. Balzano. Formulation and Characterization of Tissue Equivalent Liquids Used for RF Densitometry and Dosimetry Measurements. *IEEE Transactions on Microwave Theory and Techniques*, 52(8):2046–2056, Aug. 2004.

[56] T. Karagiannis, J.-Y. L. Boudec, and M. Vojnovic. Power-Law and Exponential Decay of Inter-Contact Times Between Mobile Devices. In *Proc. of ACM Mobicom 2007*, pages 183–194. Association for Computing Machinery, Inc., Sept. 2007.

[57] B. Krishnamurthy and C. Wills. Privacy diffusion on the web. In *Proceedings of the 18th international conference on World wide web*

- *WWW '09*, page 541, New York, New York, USA, Apr. 2009. ACM Press.

[58] Z. Le, G. Vakde, and M. Wright. PEON. In *Proceedings of the 2nd International Conference on PErvsive Technologies Related to Assistive Environments - PETRA '09*, pages 1–8, New York, New York, USA, June 2009. ACM Press.

[59] V. Lenders, G. Karlsson, and M. May. Wireless Ad Hoc Podcasting. In *SECON*, 2007.

[60] V. Lenders, J. Wagner, and M. May. Analyzing the impact of mobility in ad hoc networks. *REALMAN*, 2006.

[61] M. Li, N. Cao, S. Yu, and W. Lou. FindU: Privacy-preserving personal profile matching in mobile social networks. In *2011 Proceedings IEEE INFOCOM*, pages 2435–2443. IEEE, Apr. 2011.

[62] Y. Li, M. Qian, D. Jin, P. Hui, Z. Wang, and S. Chen. Multiple mobile data offloading through disruption tolerant networks. *IEEE Transactions on Mobile Computing*, 13(7):1579–1596, 2014.

[63] L. Lilien, Z. Kamal, V. Bhuse, and A. Gupta. Opportunistic networks: the concept and research challenges in privacy and security. *WSPWN*, 2006.

[64] A. Lindgren, A. Doria, and O. Schelén. *Service Assurance with Partial and Intermittent Resources*, volume 3126 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, Sept. 2004.

[65] A. Mtibaa, M. May, C. Diot, and M. Ammar. PeopleRank: Social Opportunistic Forwarding. In *2010 Proceedings IEEE INFOCOM*, pages 1–5. IEEE, Mar. 2010.

[66] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, mar 2003.

[67] R. Nojima and Y. Kadobayashi. Cryptographically secure bloom-filters. *Transactions on Data Privacy*, 2(2):131–139, Aug. 2009.

[68] Open Garden. Firechat App. [Online, 2015-11-14] https://opengarden.com/firechat.

[69] Open Garden. Opportunistic API. [Online, 2015-11-14] `http://opengarden.com/apisdk/`.

[70] G. Pallapa, M. Di Francescoy, and S. K. Das. Adaptive and context-aware privacy preservation schemes exploiting user interactions in pervasive environments. In *2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6. IEEE, June 2012.

[71] P. Papadimitratos, M. Poturalski, P. Schaller, P. Lafourcade, D. Basin, S. Capkun, and J.-P. Hubaux. Secure neighborhood discovery: A fundamental element for mobile ad hoc networking. *IEEE Communications Magazine*, (February):132–139, 2008.

[72] I. Parris and T. Henderson. Practical privacy-aware opportunistic networking. In *BCS-HCI Proceedings of the 25th BCS Conference on Human-Computer Interaction*, pages 553–557. British Computer Society, July 2011.

[73] I. Parris and T. Henderson. Privacy-enhanced social-network routing. *Computer Communications*, 35(1):62–74, Jan. 2012.

[74] G. Pei and T. Henderson. Validation of NS-3 802.11b PHY model. Technical report, Boeing Research and Technology, New York, USA, 2009.

[75] C. Perkins and E. Royer. Ad-hoc on-demand distance vector routing. In *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, pages 90–100, New Orleans, LA, USA, 1999.

[76] C. Phillips, D. Sicker, and D. Grunwald. A Survey of Wireless Path Loss Prediction and Coverage Mapping Methods. *IEEE Communications Surveys & Tutorials*, 15(1):255–270, Jan. 2013.

[77] A. Picu and T. Spyropoulos. Minimum Expected *-Cast Time in DTNs. In E. Altman, I. Carreras, R. E. Azouzi, E. Hart, and Y. Hayel, editors, *BIONETICS*, volume 39 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 103–116. Springer, 2009.

[78] A.-K. Pietiläinen, E. Oliver, J. LeBrun, G. Varghese, and C. Diot. MobiClique: middleware for mobile social networking. *Proceedings of the 2nd ACM workshop on Online social networks*, 127(3):49–54, 2009.

[79] M. Pitkänen, M. Conti, A. Passarella, S. Giordano, S. Trifunovic, K. Hummel, and M. May. SCAMPI : Service Platform for Social Aware Mobile and Pervasive Computing Categories and Subject Descriptors. *Workshop on Mobile Cloud Computing (MCC)*, pages 7–12, 2012.

[80] Qualcomm. LTE Direct. [Online, 2015-11-14] `http://ltedirect.qualcomm.com/`.

[81] J. Scott, J. Crowcroft, P. Hui, and C. Diot. Haggle: a Networking Architecture Designed Around Mobile Users. In *WONS 2006 : Third Annual Conference on Wireless On-demand Network Systems and Services*, pages 78–86, 2006.

[82] S. Seneviratne, A. Seneviratne, P. Mohapatra, and P.-U. Tournoux. Characterizing WiFi connection and its impact on mobile users. In *Proceedings of the 8th ACM international workshop on Wireless network testbeds, experimental evaluation & characterization - WiNTECH '13*, page 81, New York, New York, USA, Sept. 2013. ACM Press.

[83] A. Shikfa, M. Onen, and R. Molva. Privacy in Content-Based Opportunistic Networks. In *WAINA*, 2009.

[84] N. Sonnad. $30 Smartphones are here. [Online, 2015-11-14] `http://qz.com/314285/30-smartphones-are-here-and-theyre-getting-\better-every-day/`, 2014.

[85] D. Soper. Is human mobility tracking a good idea? *Communications of the ACM*, 55(4):35, apr 2012.

[86] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. pages 252–259. ACM Press, 2005.

[87] V. Srinivasan, M. Motani, and W. T. Ooi. {CRAWDAD} dataset nus/contact (v. 2006-08-01). Downloaded from http://crawdad.org/nus/contact/20060801, Aug. 2006.

[88] Statista. Number of downloads from the Apple App and Google Play stores. [Online, 2015-11-14] http://www.statista.com/chart/1109/google-play-looks-set-to-overtake-apple-s-app-store/.

[89] Statista. Number of downloads from the Apple App store. [Online, 2015-11-14] http://www.statista.com/statistics/263794/number-of-downloads-from-the-apple-app-store.

[90] Statista. Worldwide number of mobile only Internet users. [Online, 2015-11-14] http://www.statista.com/statistics/271389/number-of-worldwide-mobile-only-internet-users.

[91] The Serval Project. Reclaim your phone. [Online, 2015-11-14] http://www.servalproject.org/.

[92] S. Trifunovic, C. Anastasiades, B. Distl, and F. Legendre. PodNetSec - Secure Opportunistic Content Dissemination. In *Demo Session of ACM Mobisys, San Francisco, CA, USA*, jan 2010.

[93] S. Trifunovic, B. Distl, D. Schatzmann, and F. Legendre. Wifi-opp: ad-hoc-less opportunistic networking. In *CHANTS*, 2011.

[94] S. Trifunovic and A. Hossmann-Picu. Stalk and lie—The cost of Sybil attacks in opportunistic networks. *Computer Communications*, may 2015.

[95] S. Trifunovic, M. Kurant, K. A. Hummel, and F. Legendre. Preventing spam in opportunistic networks. *Computer Communications*, 41:31–42, Mar. 2014.

[96] C. Tuduce and T. Gross. A mobility model based on WLAN traces and its validation. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 1, pages 664–674. IEEE, Apr. 2005.

[97] Uepaa. Alpine safety App. [Online, 2015-11-14] http://www.uepaa.ch.

[98] Uepaa. p2pkit. [Online, 2015-11-14] http://www.uepaa.ch/how2p2p.

[99] US-CERT. CryptoLocker Ransomware Infections. [Online, 2015-11-14] https://www.us-cert.gov/ncas/alerts/TA13-309A, 2013.

[100] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. *Technical report number CS-200006, Duke University*, (CS-200006):1–14, 2000.

[101] A. R. von Hippel and S. O. Morgan. Dielectric Materials and Applications. *Journal of The Electrochemical Society*, 102(3):68C, 1955.

[102] Y. Wang, T.-T. Zhang, H.-Z. Li, L.-P. He, and J. Peng. Efficient Privacy Preserving Matchmaking for Mobile Social Networking against Malicious Users. In *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 609–615. IEEE, June 2012.

[103] WiFi Alliance. WiFi Aware. [Online, 2015-11-14] http://www.wi-fi.org/discover-wi-fi/wi-fi-aware.

[104] WiFi Alliance. WiFi Direct. [Online, 2015-11-14] http://www.wi-fi.org/discover-wi-fi/wi-fi-direct.

[105] L. Wu, Xintao and Ying, Xiaowei and Liu, Kun and Chen. A Survey of Privacy-Preservation of Graphs and Social Networks. In H. Aggarwal, Charu C. and Wang, editor, *Managing and Mining Graph Data*, pages 421–453. Springer US, 2010.

[106] Z. Yang, B. Zhang, J. Dai, A. C. Champion, D. Xuan, and D. Li. E-SmallTalker: A Distributed Mobile System for Social Networking in Physical Proximity. In *2010 IEEE 30th International Conference on Distributed Computing Systems*, pages 468–477. IEEE, 2010.

[107] S. Zakhary and M. Radenkovic. Utilizing social links for location privacy in opportunistic delay-tolerant networks. In *International Conference on Communications*, 2012.

Curriculum Vitae

Bernhard Distl was born in Vienna, Austria on March 20, 1980.

Educational Curriculum Vitae

2009 – 2015	**Doctor of Sciences** *Swiss Federal Institute of Technology (ETH) Zürich*
2007 – 2009	**Teaching Diploma (Didaktischer Ausweis)** *Swiss Federal Institute of Technology (ETH) Zürich*
2002 – 2007	**M. Sc. in Electrical Engineering and Information Technology** *Swiss Federal Institute of Technology (ETH) Zürich*
1999 – 2001	**Diploma IT & Management** *Kolleg Spengergasse, Vienna, Austria*
1994 – 1998	**High school** *St. Ursula, Vienna, Austria*

Working and Teaching Experience

2011 – now	**Managing Partner** *Roman Consulting & Engineering, Zürich*
2006 – 2011	**Course instructor & Security consultant** *Roman Consulting & Engineering, Zürich*
2004 – 2007	**Teaching Assistant** *Swiss Federal Institute of Technology (ETH) Zürich*