KATHOLIEKE UNIVERSITEIT
LEUVEN

Arenberg Doctoral School of Science, Engineering & Technology
Faculty of Engineering
Department of Electrical Engineering (ESAT)

Design and analysis methods for privacy technologies

Carmela Troncoso

Dissertation presented in partial
fulfillment of the requirements for
the degree of Doctor
in Electrical Engineering

April 2011

Design and analysis methods for privacy technologies

Carmela Troncoso

Jury:
Prof. Dr. Ir. Ann Haegemans, chairman
Prof. Dr. Ir. Adhemar Bultheel, acting chairman
Prof. Dr. Ir. Bart Preneel, promotor
Prof. Dr. Ir. Claudia Diaz, promotor
Prof. Dr. Ir. Geert Deconinck
Prof. Dr. Ir. Frank Piessens
Prof. Dr. Nikita Borisov
(University of Illinois at Urbana-Champaign)
Dr. George Danezis
(Microsoft Research Cambridge)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Electrical Engineering

April 2011

© Katholieke Universiteit Leuven – Faculty of Engineering
Arenbergkasteel, B-3001 Leuven-Heverlee, Belgium

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotocopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm or any other means without written permission from the publisher.

Legal depot number D/2011/7515/52
ISBN number 978-94-6018-350-8

Acknowledgements

People have always told me that writing a thesis is a solo journey. Now I am at the end of the road and when I look back there were hard moments, but I have felt everything but alone in this quest.

First of all I want to thank my advisors Bart Preneel and Claudia Diaz, who believed in me from the first day and let me know they trusted me every day of my PhD. I would like to thank Bart for his support; for (almost) never saying no to anything that I asked for no matter how crazy the idea was (and I have had a fair amount of crazy ideas); and for telling me something that changed my life: "Shit happens, you'll deal with it". Thanking Claudia for all that she did for me in a few words is very difficult. There are many good advisors out there able to teach their students how to do research, but friends that feel like family there are very few. I am so lucky that I found both of them in one person.

I am also very grateful to Prof. Frank Piessens, Prof. Geert Deconinck, Prof. Nikita Borisov and Dr. George Danezis for serving as jury members, and giving me many suggestions to improve this dissertation. I want to thank Prof. Ann Haegemans and Prof. Adhemar Bultheel for chairing the jury.

I would not have written this thesis if I had not had the luck of sharing an office not only with Claudia, but also with George. When I knew nothing about security nor privacy he took his time to guide me, always giving me new opportunities to learn. I also thank him for introducing me to the fascinating world of Bayesian inference: the gift that keeps on giving, the Angry Nights, the Rollerkillers, the Filthy Dragon, Indigo's bagels, for teaching me my first words in Greek,…but above everything I would like to thank him for making me realize why it is worthy to work on privacy. George, I shall not forget: "Girls come and go, traffic analysis is forever not for Christmas."

I would also like to thank all of my co-authors for having the patience to work with me, letting me learn from them, and having made each new paper not only a nice academic experience but also fun. A very special thanks goes to Benedikt, a friend inside and outside of ESAT, always there to support me not only at work but also in life making his home my home (with better breakfast). I cannot imagine my life in Leuven without sharing dinner, a washing machine, a car, a bbq, garden furniture,... with him. Dude, when are you coming this summer?

Not only my co-authors have had an influence on my research. A big thanks goes to all the people that have been part of Cosic during these years and created the best working environment for me. In particular thanks to Josep, the best office mate ever; to Seda and Danny, always eager to help and take care of the little things that make my life so much easier; to Pela, that made the damn paperwork even look nice when I was having a coffee at her desk; and to Fonsi, Orr, Markulf, Elena, Basti, Emily, Kåre, Ero, for so many coffees, laughters, beers, dinners, game evenings, climbing hours, etc. I would also like to thank my office and lunch mates in Vigo for making going to work a pleasant experience in my last months of writing.

A more than special thanks go to Elketje. I would need to write a whole new book to really express all that I am thankful for, but in few words: thank you not only for letting me see the different sides of Elke, and teach me that there is more than one point of view in life and thus there is not only one definition for right and wrong; but more than anything for being there every day to remind me.

Then there are all those people that do not know a word about security or maths, but that have been as important for my PhD as those whom I worked with. The warmest thanks to Alba and OneC de Montalvo, for never letting me give up, for making pancakes when they were needed, for understanding how cool it would be to introduce a car in a DVD, and for always being ready to give up a bit of their happiness to help me.

Thanks to Mil Amigos, each and one of them, because no matter how far they are they always sound near and with them I always know that whatever it is, it is going to be “chichi!”. Thanks to Fra and Sarah for reminding me I actually enjoy going out, and having a life. I would never have written this thesis without Haydée and Who, always there to calm me down on the other side of the phone. And a very special thanks to Eva, for always encouraging me to follow my dreams even when they were not aligned with hers. I would not have a PhD without her.

Finally, I am deeply thankful to my family. Thanks to my parents for building a safety net such that I could run around catching dreams without fear to fall, and to my sister for being green and listen to me and always being so keen in telling me I am overreacting. I also thank my aunt Arita, my godmother Celia, and the whole Quiroga clan for always have their door open to me, ready to give me exactly what I need without ever making questions.

To all those I have mentioned, and all those that are not mentioned but have been by my side all these years:

Thank you for being a part of this!

I would like to acknowledge the K.U.Leuven, the Fundación Pedro Barrié de la Maza, and the Fund for Scientific Research in Flanders (FWO) for funding my research.

Abstract

As advances in technology increase data processing and storage capabilities, the collection of massive amounts of electronic data raises new challenging privacy concerns. Hence, it is essential that system designers consider privacy requirements and have appropriate tools to analyze the privacy properties offered by new designs. Nevertheless, the privacy community has not yet developed a general methodology that allows engineers to embed privacy-preserving mechanisms in their designs, and test their efficacy. Instead, privacy-preserving solutions are designed and analyzed in an *ad hoc* manner, and hence it is difficult to compare and combine them in real-world solutions.

In this thesis we investigate whether general methodologies for the design and analysis of privacy-preserving systems can be developed. Our goal is to lay down the foundations for a privacy engineering discipline that provides system designers with tools to build robust privacy-preserving systems.

We first present a general method to quantify information leaks in any privacy-preserving design that can be modeled probabilistically. This method allows the designer to evaluate the degree of privacy protection provided by the system. Using anonymous communication systems as case study we find that Bayesian inference and the associated Markov Chain Monte Carlo sampling techniques form an appropriate framework to evaluate the resistance of these systems to traffic analysis. The Bayesian approach provides the analyst with a neat procedure to follow, starting with the definition of a probabilistic model that is inverted and sampled to estimate quantities of interest. Further, the analysis methodology is not limited to specific quantities such as “who is the most likely receiver of Alice’s message?,” but can be used to answer arbitrary questions about the entities in the system. Finally, our methodology ensures that systematic biases in information analysis are avoided and provides accurate error estimates.

In the second part of this thesis we tackle the design of privacy-preserving systems, using pay-as-you-drive applications as case study. We propose two pay-as-you-drive architectures that preserve privacy by processing personal data locally to the users, and only communicating billing information to the provider.

Local processing enhances privacy, but may be detrimental to other security properties such as service integrity (e.g., the provider has access to less data when verifying the correctness of the bill). We design a protocol that, using advanced cryptographic primitives, allows users to prove to the service provider that they have correctly performed the computation, while revealing the minimum amount of location data. Finally, our designs are validated from a security, performance and legal perspective, to ensure that they are ready for deployment.

Based on the lessons learned while designing privacy-preserving schemes for pay-as-you-drive applications, we identify the basic steps to be performed when designing new privacy-preserving solutions that minimize the disclosure of personal data while fulfilling other essential security requirements. We argue that, first of all, the designer must explicitly identify the basic functionality of the system, and the minimum set of data that needs to be revealed to service providers. Then, multi-lateral security requirements have to be addressed and protective measures are established to safeguard the interest of all entities in the system while enabling users to disclose a minimum amount of personal information. Even though in this thesis we use pay-as-you-drive applications as a central case study, the general applicability of these steps has been tested in the design of a privacy-preserving e-petition system, in which user's privacy is guaranteed by hiding their identity from the provider while revealing their preferences.

Samenvatting

De technologische vooruitgang maakt het mogelijk om steeds meer informatie te verwerken en op te slaan. Het verzamelen van massale hoeveelheden elektronische informatie creëert nieuwe en uitdagende privacybekommernissen. Het is daarom belangrijk dat systeemontwerpers rekening houden met privacyvereisten en dat ze de geschikte werkmiddelen voorhanden hebben om de privacyeigenschappen van nieuwe ontwerpen te analyseren. De privacygemeenschap heeft evenwel nog geen algemene methodologie ontworpen die ontwerpers toelaat om privacybehoudende mechanismen in hun ontwerpen te verwerken en om de doeltreffendheid ervan te testen. In de plaats daarvan worden privacybehoudende oplossingen *ad hoc* ontworpen en geanalyseerd, en is het daardoor moeilijk om ze te vergelijken en met elkaar te combineren in echte producten.

In dit proefschrift onderzoeken we of algemene methodologieën kunnen ontwikkeld worden om privacybehoudende systemen te ontwerpen en te analyseren. Ons doel is de grondbeginselen van een privacyontwerpdiscipline uiteen te zetten die het systeemontwerpers mogelijk maakt om met geschikte hulpmiddelen robuuste privacybehoudende systemen te bouwen.

Allereerst stellen we een algemene methode voor om informatielekken te kwantificeren in elk privacybehoudend ontwerp dat probabilistisch kan gemodelleerd worden. Deze methode laat de systeemontwerper toe te evalueren in welke mate het systeem privacybeschermend is. Door anonieme communicatiesystemen als voorbeeld te bestuderen, komen we tot de conclusie dat Bayesiaanse statistiek en de bijhorende Markovketen-Monte Carlo-experimenttechnieken een geschikt kader vormen om te evalueren of deze systemen bestand zijn tegen traffiekanalyse. De Bayesiaanse aanpak geeft de analyst een welgedefinieerde procedure die hij kan volgen, uitgaande van de definitie van een probabilistisch model dat wordt geïnverteerd en getoetst om de interessante grootheden te schatten. Verder beperkt de analysemethode zich niet tot specifieke vragen zoals ”wie is de meest aannemelijke ontvanger van de boodschap van Alice?”, maar kan ze ook gebruikt worden om arbitraire vragen over de entiteiten in het systeem te beantwoorden. Tenslotte verzekert onze methode dat systematische vertekeningen

in de informatieanalyse vermeden worden en levert het nauwkeurige schattingen van foutenmarges op.

In het tweede deel van dit proefschrift behandelen we het ontwerpen van privacybehoudende systemen, toegepast op tolrijden. We stellen twee tolrijarchitecturen voor die de privacy van persoonsgegevens waarborgen door persoonlijke data lokaal te verwerken en enkel facturatiegegevens naar de dienstverlener door te sturen. De lokale verwerking verhoogt de privacy, maar kan nadelig zijn voor andere beveiligingseigenschappen zoals de integriteit van de dienstverlening (b.v., de dienstverlener krijgt minder informatie ter beschikking om de correctheid van de rekening te verifiëren). We ontwerpen een protocol dat de gebruikers ervan toelaat met geavanceerde cryptografische primitieven aan de dienstverlener te bewijzen dat ze de berekeningen correct hebben uitgevoerd, terwijl ze een minimum aan locatiegegevens vrijgeven. Tenslotte hebben we onze ontwerpen gevalideerd vanuit een beveiligings-, performantie- en legaal standpunt om te garanderen dat ze klaar zijn om in gebruik te nemen.

Op basis van de lessen die we geleerd hebben tijdens het ontwerpen van privacybehoudende systemen voor tolrijtoepassingen, hebben we de basisstappen geïdentificeerd die moeten worden uitgevoerd bij het ontwerpen van nieuwe privacybehoudende oplossingen die het aantal persoonsgegevens dat moet onthuld worden minimaliseert, en die toch geen afbreuk doen aan andere essentiële beveiligingsvereisten. We geven aan dat de ontwerper van een systeem in de eerste plaats de basisfunctionaliteit ervan expliciet moet identificeren, tesamen met de kleinste hoeveelheid informatie die kan ter beschikking gesteld worden aan de diensverleners. Pas daarna kunnen andere beveiligingsvereisten in rekening gebracht worden en kunnen beschermingsmaatregelen worden opgesteld om de belangen van alle partijen in het systeem te waarborgen en de gebruikers ervan in staat te stellen om slechts een minimum aan persoonlijke informatie vrij te geven. Hoewel we in dit proefschrift tolrijtoepassingen als centraal thema gebruiken, wordt de algemene toepasbaarheid van deze stappen aangetoond bij het ontwerp van een privacybehoudend ePetitiesysteem waarbij de gebruikersprivacy gegarandeerd wordt door hun identiteit te verbergen voor de dienstverlener wanneer hun voorkeur onthuld worden.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

New communication technologies, such as the Internet, mobile phones, Bluetooth, etc. enable a broad range of interactive applications, multimedia services and pervasive communications for consumers. These services are radically changing society, in the sense that they are affecting the way people interact with each other and with institutions (e.g., banks, government, etc.). Electronic communications facilitate the realization of an increasing number of activities providing better or more accurate information, and more comfortably, for the user. As a result, interaction amongst humans is progressively being mediated, or even substituted, by interaction with machines. For instance, people invite friends to events using Facebook, share their vacation photos using Google's Picasa albums, query Google Maps from a smart phone instead of asking strangers for directions, or use online banking from their homes instead of going to the bank to check their account's balance.

The use of electronic communications changes the flow of information with respect to an off-line society in terms of data storage, data distribution, ease of access to data, etc. This new flow, together with an increased availability of information, raises new concerns and risks with respect to privacy [58,135,203,253]. In the off-line world we have established strategies to protect our private information. We are very selective about what we tell to whom, where, and when in order to control what others think of us, or to protect our safety. Daily examples of this include employees not sharing their whereabouts on a Saturday night with their employers or colleagues, or people not broadcasting that their houses will be empty during vacation. Off-line, physical constraints eased the control of personal information;

for example, walls protected private conversations from being overheard; or paper-based communications and databases hindered automated data mining, and they prevented information from being trivially copied and hence rapidly spread.

In the online world, however, electronic communications have drastically changed the situation. Nowadays information can be easily collected, aggregated, analyzed, copied, and exported to other contexts. In the past people's whereabouts could only be obtained by means of expensive physical surveillance, while currently they can be obtained (remotely and at practically no cost) from voluntary disclosure on the Internet [1, 98]; or they can be inferred from online activity records of social networks, instant messaging, web forums, etc. These inferences become more effective when combined with mobile communications that reveal people's location as demonstrated by Krumm [167], or Golle and Partridge [127]. Another danger related to the easy collection of information is the existence of automated advanced mining techniques, that allow for massive profiling of both individuals and communities [58, 179].

Information in physical form (e.g., microfilm, paper, etc.) is easy to delete once it is no longer necessary: it suffices to destroy the physical container. This is not the case for electronic data due to the problem of data remanence [125] (i.e., residual data remains recoverable even after attempts have been made to remove or erase these data). Sometimes, even destroying the physical container of electronic data is not enough to guarantee that these data cannot be recovered [219]. Furthermore, even when it can be ensured that data has been deleted in a single place, the ease with which these data can be copied and distributed makes it extremely hard to guarantee that the deletion has been effective, as copies could exist somewhere else (e.g., backups). This means that digital traces are likely to have a longer life-time than expected at creation, or even to be never forgotten. To illustrate the downside of such an environment let us consider a blog in which political opinions are shared. Even if people's ideas change over time a certain opinion published in the past is likely to be registered forever (at the service provider hosting the blog or any other web pages where the opinion was duplicated) and possibly available to search engines. This information jeopardizes people's right to re-define their identity without being permanently marked by actions from their past [34].

The research community has tackled privacy problems from different perspectives, developing a wide range of solutions that achieve different privacy properties such as anonymity (a subject is anonymous if she is not identifiable within a set of subjects, the anonymity set), unlinkability (two actions are unlinkable if they are no more and no less related than they are related concerning any a priori knowledge), or unobservability (an action is unobservable if performing it is indistinguishable from not performing any action at all). We refer the interested reader to the terminology by Pfitzmann and Hansen [217] for further definitions of privacy properties in a computer science context; and to [134] for a discussion on how the definitions of these properties vary in social, academic and legal contexts.

In this thesis we concentrate on technical privacy solutions developed by computer scientists and engineers. These solutions can be classified into two main categories depending the privacy model considered in their design.

In the first category the privacy model assumes that users reveal their private data to *trusted* data controllers (e.g., service providers). Data controllers are in charge of protecting privacy against external parties such as eavesdroppers, hackers, malicious insiders, etc. These solutions rely on protection means such as security policies or access control mechanisms. Further, system audits are used to detect whether something goes wrong, and to find liable parties when a privacy breach occurs. Consider for example technologies that support Data Protection compliance according to the European Data Protection Directive [95]. This directive focuses on ensuring that users consent to the collection of data, and that this collection is proportionate according to the provision of the service forbidding further processing and transfer of the collected data.[1] When these conditions are not met the data controller may be held liable. For instance, when data is collected without the consent of the users, the data collected is excessive for the purpose of the application (e.g., asking customers about their marital status when buying goods from the Internet), or the data is processed for unauthorized secondary uses (i.e., use of data for purposes other than those for which it was originally collected).

These solutions only offer "soft" privacy guarantees to users, in the sense that once a user reveals her data to a trusted service provider she has little control on how these data are later processed or shared. At this point, whether privacy is protected or not depends on the trustworthiness of the controller, as well as her competence when handling users' data. We discuss further disadvantages of "soft" privacy solutions in [133].

In the second category, instead of relying on an trusted service provider to protect their data, the model assumes that users actively take part in protecting their privacy by using so-called Privacy Enhancing Technologies (PETs). PETs are technical measures to protect privacy by eliminating or minimizing the disclosure of personal data, hence preventing unnecessary or unwanted processing of personal data, without the loss of the functionality of the information system [274]. An example of a PET are anonymous credentials [43], that allow users to prove that a certain statement is true without revealing any further information. For instance, a credential may contain someone's address signed by a certification authority. This credential can be used to prove that the subject lives in a given neighborhood without revealing the particular street, house number, or any other information. Another example is Private Information Retrieval [171] that allows authorized users to query a database without revealing to the database owner which items have been retrieved. These solutions offer "hard" privacy protection to their users

[1] There are exceptions for this prohibition when it comes to the processing of data for historical, statistical or scientific purposes [95].

by providing them with means to minimize the need to trust any entity in the system with the protection and control of their personal data. It must be noted that PETs come at a cost, as in general they require more computational and/or communication resources than privacy-invasive technologies, but on the other hand minimizing the data collected reduces the maintenance costs of the service provider as the system now handles less personal data.

Although PETs limit the amount of data disclosed, their use does not preclude “soft” privacy protection. For some applications the collection of personal data is unavoidable and the data controller must put in place adequate mechanisms (privacy policies, access control, etc.) to safeguard the collected personal information. As an example, let us consider thee pay-as-you-drive applications described in the second part of this thesis. As we will see, in these applications, in order to provide the pay-as-you-drive service, the provider only needs to know the final fee users must pay, but not their detailed driving record. Yet, the provider has financial information about the users, and this information must be kept confidential towards unauthorized parties.

Designing systems that give hard privacy guarantees is a non-trivial process in which privacy, as well as other security requirements (confidentiality, integrity, availability, etc.), have to be fulfilled. As new applications that introduce new privacy and security concerns arise, new architectures, cryptographic primitives and protocols need to be developed that address these concerns. Once the system is designed, analyzing it to ensure that indeed no information is inadvertently revealed is also an arduous task.

1.2 This thesis

In this thesis we tackle the problem of designing systems that provide hard privacy guarantees to their users. We aim to lay down the foundations for a privacy engineering discipline that provides system designers with tools to build robust privacy-preserving systems. These tools shall allow engineers to embed privacy-preserving mechanisms in their designs, and test them for potential information leaks – assuming that such a leakage could lead to a privacy breach.

This thesis is divided into two parts. In the first part we deal with the analysis of privacy-preserving systems, and in the second part we study how to design privacy-preserving systems. We now provide a short introduction to these topics and outline the contributions that can be found in each part.

1.2.1 Analysis of privacy-preserving systems

The first part of this thesis investigates whether there is a general way to quantify information leaks in any privacy-preserving design such that we can evaluate the design's degree of protection. We choose anonymous communication systems as a case study to illustrate the difficulty of quantifying the amount of leaked information, and how this leakage affects the privacy properties of the system. As stated by Diffie and Landau [90]: "Communication is fundamental to our species; private communication is fundamental to both our national security and our democracy." Protecting the content of messages is not enough to provide private communications. An attacker can exploit traffic information such as duration, frequency, origin, or destination of a communication to infer facts about ongoing communications even if the content of messages is encrypted. The goal of anonymous communications is to thwart traffic analysis and conceal who speaks to whom.

Since Chaum proposed the first system for anonymous email in 1981 [47] there have been numerous proposals of systems that conceal the identity of the source and/or recipient of a communication [28,72,123,185,187,189,192,199,215,227]. The robustness of these anonymous communications systems is analyzed and improved using adversary models and attacks to prove the absence or presence of unwanted leakages. The vast majority of these solutions still leak information that can be exploited by an adversary who observes the system. From these observations, the adversary may be able to infer who communicates with whom, or other characteristics of the communication. In general, these attacks are enabled by information leaks that had been unforeseen at the design stage, or leaks that are inevitable to achieve efficiency.

We first present a simple attack that relies on classic optimization techniques to uncover persistent patterns of communication. The attack is better at de-anonymizing communications than previous proposals because it considers all users at once, rather than single users iteratively. It is however computationally limited to rather small and simple systems. We then propose a novel analysis methodology based on Bayesian inference that incorporates all aspects of a system likely to reveal information such as path building constraints [238], route fingerprinting [77], user behavior [89], etc. Further, these Bayesian techniques are based on sampling, hence they do not suffer from the computational limitations of previously proposed methods and can be used to efficiently analyze complex systems.

Bayesian techniques provide a sound framework to analyze systems and co-estimate multiple quantities. Contrary to previous proposals for traffic analysis, the output of our method does not correspond to a specific inference, such as "who is the most likely receiver of Alice's message?," but can be used to infer arbitrary statements such as "has Alice ever communicated with Bob," "is Alice better friends with Bob or Charlie," or "are Alice's two best friends Debbie and Emily?" Besides their flexibility, Bayesian techniques provide reliable error estimates, allowing the analyst studying the system to evaluate the confidence she can have on the inferences.

In this thesis we use mix networks [72, 192] to illustrate how our Bayesian framework operates. Nevertheless, we note that the framework can accommodate any type of anonymous communication network, or other privacy-preserving systems, and evaluate their resistance to traffic analysis. We hope that our work serves as a starting point for the creation of standard analysis methodologies to evaluate information leakage in privacy-preserving designs.

1.2.2 Design of privacy-preserving systems

Designing a privacy-preserving system is a complex task. As previously discussed, encrypting the content of communications and trusting data controllers for protecting information is insufficient to protect users' privacy. A first, straightforward, approach to ensure privacy could be to not give any private information to the service provider. However, hiding *all* private information from providers can be detrimental for the service because it may conflict with other security requirements of the system, such as integrity or accountability; and in some extreme cases even prevent the provision of the service itself. To illustrate this problem let us consider an electronic petition system[2] in which citizens can sign formal requests addressed to an authority. For this purpose, identifiability of signers is not strictly necessary: what is important is how many people support a petition; not who they are. Given that petitions may encode personal information (e.g., religious views, political opinion, etc.) it seems appropriate to allow the signers to be anonymous. Nevertheless if enabling anonymous signatures prevents the e-petition server from detecting that a user has signed a petition multiple times, then the result of the petition becomes meaningless. We note that e-petition systems in which privacy and security requirements are satisfied simultaneously can be built [83].

The second part of this thesis deals with the design of privacy-preserving systems. We choose pay-as-you-drive (PAYD) applications to illustrate the feasibility of building privacy-preserving applications that satisfy other security requirements, while being ready for real-world deployment.

Road taxes, and vehicle insurance, are usually associated with a flat rate paid monthly or yearly by drivers. In PAYD schemes, in contrast, users are charged a personalized fee based on their driving records. PAYD systems require that users' identity is revealed for accountability and billing purposes. Therefore, the anonymous communications-based solutions we discussed in the first part of this

[2] `http://epetitions.net/` or `http://www.public-i.info/products/epetitions/`

thesis are not applicable. We must find other ways for safeguarding privacy and minimize the disclosure of personal data, other than users' identity, to service providers.

Current implementations of such systems rely on sending fine-grained location data to service providers [53, 54, 94, 208]. This creates a privacy risk for drivers because the trajectories followed by an individual may reveal sensitive information such as health status, political affiliation, or religious beliefs thus jeopardizing location privacy. Beresford and Stajano [26] define location privacy as "the ability to prevent other parties from learning one's current or past location;" while Duckham and Kulik [99] refine the concept of location privacy by defining it as "a special type of information privacy which concerns the claim of individuals to determine for themselves when, how, and to what extent location information about them is communicated to others." For instance, a person frequently visiting an oncology clinic exposes her medical condition. Similarly, location records may reveal that a user regularly visits the Republican party or the Democratic party headquarters, hence disclosing the user's political views. Moreover, fine-grained location information is not strictly necessary for the provision of the service, namely *charging users depending on their driving behavior.* In fact, PAYD systems must only comply with two simple requirements: i) the provider must know the final fee to charge; ii) the provider must be convinced that this fee is correctly computed and users cannot undetectably commit fraud.

We present two architectures, PriPAYD and PrETP, in which users enjoy the advantages of PAYD while preserving their location privacy. In both systems the design principle is that personal information must be processed under the control of the user, instead of outsourced to a service provider. This is desirable from a privacy point of view, but in principle it makes it more difficult to protect the interest of service providers and verify that users have not tampered with the system. Our designs ensure that minimal personal information is revealed, hence maximizing the privacy protection of users, while fulfilling the application's integrity and accountability requirements.

Our first design, PriPAYD, demonstrates that PAYD fees can be computed without revealing fine-grained location data to the service provider, thus providing strong privacy guarantees. Our second design, PrETP, illustrates how the system's security requirements can be fulfilled at the same time. PrETP combines local computations and advanced cryptography to achieve hard privacy, integrity, and accountability. We design a novel cryptographic protocol, Optimistic Payment, that allows the service provider to verify that users pay the correct fee according to their road usage, without revealing detailed location records.[3] This way users can prove that they use genuine data and perform correct operations while disclosing the minimum amount of location data.

[3] We outline other mechanisms that do not rely on cryptography to verify that users do not misbehave in [267].

In order to evaluate and complete our design we thoroughly analyze our system from a security, legal,[4] and performance perspective. We prove Optimistic Payment secure under standard assumptions, and demonstrate that its use fulfills the security requirements of the system. Further, we provide an efficient implementation of our protocol on an embedded microcontroller ready to use on vehicles, and prove that the back-end server can be constructed with off-the-shelf technology.

Finally, we revisit our design and identify general principles that allow to design solutions with embedded hard privacy-preserving guarantees. These design principles are applicable to many other applications, as for instance Smart Energy systems [228]. It is our hope that the steps taken in our design process, further elaborated in [133], guide the design of future privacy-preserving systems.

1.3 Outline and summary of contributions

In this section we give an overview of the outline of this thesis and summarize the contributions that can be found in each chapter. Our goal is to facilitate the evaluation of the overall contribution of this dissertation. The content of Chapters 3, 4, 5, and 7 has been published in the proceedings of peer-reviewed international conferences and journals [17, 79, 133, 266–268, 270].

Part I: Analysis of Privacy-preserving systems

Chapter 2: Traffic Analysis in Anonymous Communications. We introduce traffic analysis, and survey several techniques that have been proposed to analyze Anonymous Communications systems from the start of the field in 1981 until today. This chapter aims at giving context to the methodologies for analyzing anonymous communication systems presented in Chapters 3 to 5.

Chapter 3: Perfect Matching Disclosure Attacks. We show that previous attacks to uncover the identity of users that communicate repeatedly through an anonymous communications system may not work in practice because of simplistic assumptions about user behavior. We first define a non-restrictive user behavior model, in which users have an arbitrary number of friends and arbitrary preferences towards them. Then, we propose an attack based on finding perfect matchings amongst senders and receivers of messages over a mix network. This attack outperforms previous proposals. Finally, we introduce an enhanced profiling methodology that recovers users' profiles more accurately than its predecessors.

[4] For the sake of brevity the legal analysis of PriPAYD and PrETP has been left out of this thesis. It can be found in [16, 17, 267, 268].

This work was published in [270] and it is joint work with Benedikt Gierlichs, Bart Preneel, and Ingrid Verbauwhede. I have provided most of the key ideas. The modeling and simulation workload was shared amongst the authors.

Chapter 4: Bayesian Inference to De-anonymize Persistent Communications. We re-examine the problem of extracting communication profiles and revealing communication partners. Casting the de-anonymization problem as an inference problem, we use modern Bayesian statistics to simultaneously infer profiles and uncover who speaks with whom. The contributions of this chapter are: a very general model to represent long-term attacks against arbitrary anonymity systems; and the application of Bayesian inference techniques to traffic analysis of anonymous communications.

This work was published in [79] and it is joint work with George Danezis. The work was shared between the co-authors.

Chapter 5: A Bayesian Framework for the Analysis of Anonymous Communication Systems. We introduce a framework able to accommodate most attacks on anonymous communications systems proposed in the literature. This framework provides an estimation of the probability distributions necessary for computing a wide variety of anonymity metrics for relay-based mix networks. Using Bayesian inference techniques we are able to analyze systems with arbitrarily complex constraints and sizes for the first time. Our technique has several advantages over previous analysis methods. First, it optimally uses all information leaked by the system. Second, it obtains the posterior probability over all scenarios of interest, as opposed to previous attacks that only provided most likely candidates. Finally, it provides accurate error estimates useful to assess the confidence one can put in the obtained results.

This work was published in [266] and it is joint work with George Danezis. The work was shared between the co-authors.

Part II: Design of Privacy-preserving systems

Chapter 6: Location Privacy: an Overview. We introduce location privacy and give an overview of different techniques to preserve location privacy proposed in the literature. This chapter provides the context in which the privacy-preserving solutions presented in the next chapter were developed.

Chapter 7: Privacy-Friendly Electronic Road Pricing Applications. We propose two privacy-preserving solutions for pay-as-you-drive applications. Our first contribution is a decentralized architecture that protects the privacy of drivers by keeping sensitive data at the client side. Secondly, we introduce a cryptographic protocol, Optimistic Payment, that allows users to prove their honesty when computing the fee they must pay while revealing the minimal amount of location data. We define our protocol in the ideal-world/real-world paradigm, provide a construction, and prove it secure under standard assumptions. We also provide an efficient implementation of this protocol in an embedded microcontroller. Our prototype proves that our scheme is suitable for real-world deployment. Finally, we revisit the design decisions taken while designing the proposed solutions and identify common tasks and activities that can be used by engineers when designing privacy-preserving systems.

This work was published in [17, 133, 267, 268] and it is joint work with Josep Balasch, George Danezis, Claudia Diaz, Christophe Geuens, Seda Gürses, Eleni Kosta, Bart Preneel, Alfredo Rial, and Ingrid Verbauwhede.
I am the main contributor of [267,268] including the writing of the text, in [133] I contributed with one of the use cases and the distillation of the design methodology, and in [17] the work was shared amongst the co-authors.

Chapter 8: Conclusions and future work. Finally, we draw conclusions and discuss future lines of research to continue our work.

1.4 Other contributions

The content included in this dissertation is a selection of the contributions we have made to the field of privacy, according to their relevance for the topic we investigate in this thesis. Our other contributions, which have been published at several peer-reviewed conferences, have not been included here in order to maintain a reasonable thesis length. Nevertheless, these findings have influenced the results presented in this thesis in one way or another. In this section we shortly summarize these publications. The publications are classified according to their main topic and presented in inverse chronological order.

Anonymity metrics.

Revisiting A Combinatorial Approach Toward Measuring Anonymity. We identify a flaw in the "system's anonymity level" proposed by Edman *et al.* [103], which is a combinatorial approach to measure the anonymity provided by a system as a whole. This metric is based on the number of possible bijective mappings between the inputs and the outputs of a mix. We show that the "system's anonymity level" fails to capture the anonymity loss caused by subjects sending or receiving more than one message. We generalize the metric from scenarios in which user relations can be modeled as yes/no relations to cases where subjects send and receive an arbitrary number of messages.

This work was published in [121] and it is joint work with Benedikt Gierlichs, Claudia Diaz, Bart Preneel, and Ingrid Verbauwhede. The work was shared amongst the co-authors.

On the Impact of Social Network Profiling on Anonymity. The contribution of this paper is twofold. First, we propose a Bayesian method to combine multiple available sources of information and obtain an overall measure of anonymity. Second, we consider adversary models in which the attacker has incomplete or erroneous prior information. We characterize the adversary's knowledge of the social network by its quantity, quality and depth; and discuss the implications of these properties for anonymity.

This work was published in [89] and it is joint work with Claudia Diaz and Andrei Serjantov. I contributed with the evaluation and simulation of the proposed method.

Does Additional Information Always Reduce Anonymity? We identify a common misconception: the entropy of the probability distribution identifying potential senders or receivers of a message does not always decrease given more information. We show the relation of these a posteriori distributions with the Shannon conditional entropy, which is an average over all possible scenarios.

This work was published in [87] and it is joint work with Claudia Diaz and George Danezis. The work was shared amongst the co-authors.

Analysis of anonymous communications systems.

On the Difficulty of Achieving Anonymity for Vehicle-2-X Communication. Vehicle-2-X communications are hailed as the future to improve safety on the roads. Ensuring that messages sent by vehicles contain correct information is crucial to fulfill this objective, as misleading information could disrupt traffic and create potentially dangerous situations. In this work we analyze two solutions for anonymous authentication, proposed by IntelliDrive, US Department of Transportation (DoT), that trade off privacy and efficiency [273]. We show that by exploiting the reuse of pseudonyms and spatio-temporal constraints the service provider is capable of tracking a large percentage of vehicles. Furthermore, we find that one of the schemes fails to provide privacy even if the adversary does not control the service provider and only listens to the communications of vehicles.

This work was published in [56] and it is joint work with Enrique Costa-Montenegro, Stefan Schiffner, and Claudia Diaz. I provided most of the key ideas for the analysis.

Impact of Network Topology on Anonymity and Overhead in Low-Latency Anonymity Networks. We study the trade-off between anonymity and overhead for low-latency anonymity networks when dependent link padding is used. Our main finding is that the choice of the network topology has an important influence on the padding overhead and the level of anonymity provided. We consider three topologies in our study: free routes, cascades and stratified networks. We find that Free routes become impractical due to feedback effects that induce disproportionate amounts of padding. Cascades have lowest padding overhead at the cost of poor scalability with respect to anonymity. Finally Stratified networks offer the best trade-off.

This work was published in [84] and it is joint work with Claudia Diaz and Steven J. Murdoch. I contributed with the evaluation and simulation of the proposed scheme.

The Wisdom of Crowds: Attacks and Optimal Constructions. We present a traffic analysis of the ADU anonymity scheme [193]. Our results show that the quest for improving Crowds [226] is bound to fail, since we prove that the original Crowds routing algorithm provides the best security for any given mean messaging latency. Additionally we present D-Crowds, a scheme that supports any path length distribution, while leaking the least possible information, and quantify the optimal attacks against it.

This work was published in [68] and it is joint work with George Danezis, Claudia Diaz, and Emilia Käsper. The work was shared amongst the co-authors.

Two-sided Statistical Disclosure Attack. We introduce a new traffic analysis attack to uncover the receivers of messages sent through an anonymizing network supporting anonymous replies. We show that the Two-sided Statistical Disclosure Attack is superior to previous attacks when replies are routed in the system.

This work was published in [70] and it is joint work with George Danezis and Claudia Diaz. The work was shared amongst the co-authors.

Design of privacy-preserving systems.

Scalable Anonymous Communication with Provable Security. We explore new primitives for scalable anonymous communication, with a focus on providing provable security guarantees. First, we propose a new approach for secure and anonymous peer-to-peer communications based on a reciprocal neighbor policy. Secondly, we propose PIR-Tor, a centralized scalable architecture for anonymous communications based on Private Information Retrieval.

This work was published in [190] and it is joint work with Prateek Mittal, Nikita Borisov, and Alfredo Rial. I contributed with the second proposed scheme, its *simulation and evaluation.*

Drac: An Architecture for Anonymous Low-Volume Communications. We present Drac, a system designed to provide anonymity and unobservability for real-time instant messaging and voice-over-IP communications against a global passive adversary. The system uses a relay-based anonymization mechanism where circuits are routed over a social network in a peer-to-peer fashion, using full padding strategies and separate epochs to hide connection and disconnection events.

This work was published in [71] and it is joint work with George Danezis, Claudia Diaz, and Ben Laurie. The work was shared amongst the co-authors.

Efficient Negative Databases from Cryptographic Hash Functions. A negative database is a privacy-preserving storage system that allows to efficiently test whether an entry is present, but makes it hard to enumerate all entries. In this paper we propose a construction for negative databases reducible to the security of well-understood primitives, such as cryptographic hash functions or the hardness of the Discrete-Logarithm problem. Our constructions require only $\mathcal{O}(m)$ storage in the number m of entries in the database, and linear query time. These performance values improve significantly over previous work [109].

This work was published in [67] and it is joint work with George Danezis, Claudia Diaz, Sebastian Faust, Emilia Käsper, and Bart Preneel. The work was shared amongst the co-authors.

Location privacy.

Unraveling an Old Cloak: k-anonymity for Location Privacy. This paper analyzes the effectiveness of k-anonymity-based solutions for protecting location privacy and shows that these approaches have fundamental flaws. First, we identify an inconsistency between the cloaking mechanism and the k-anonymity metric. Second, while a query can be k-anonymous via cloaking, this does not necessarily protect users' location privacy. We conclude that the inconsistencies of the k-anonymity metric with respect to users' actual location privacy makes k-anonymity schemes unreliable and ineffective for location privacy.

This work was published in [248] and it is joint work with Reza Shokri, Claudia Diaz, Julien Freudiger, and Jean-Pierre Hubaux. The work was shared amongst the co-authors.

Steganographic file systems.

A Framework for the Analysis of Mix-Based Steganographic File Systems. The goal of Steganographic File Systems (SFSs) is to protect users from coercion attacks by providing plausible deniability on the existence of hidden files. We consider an adversary who can monitor changes in the file store and use this information to look for hidden files when coercing the user. We outline a high-level SFS architecture that uses a local mix to relocate files in the remote store, and thus prevent traffic analysis attacks [269] that rely on low-entropy relocations. We define probabilistic metrics for unobservability and (plausible) deniability, and present an analytical framework to extract evidence of hidden files from the adversary's observation (before and after coercion).

This work was published in [88] and it is joint work with Claudia Diaz and Bart Preneel. I contributed with the evaluation and simulation of the proposed scheme.

Traffic Analysis Attacks on a Continuous-Observable Steganographic File System. We present two attacks on the continuously-observable steganographic file system proposed by Zhou, Pang and Tan [294], which is claimed to provide provable security against traffic analysis. Our attacks are highly effective in detecting file updates and revealing the existence and location of files. Our results suggest that simple randomization techniques are not sufficient to protect steganographic file systems from traffic analysis attacks.

This work was published in [269] and it is joint work with Claudia Diaz, Orr Dunkelman, and Bart Preneel. I provided most of the key ideas. I did all the modeling and simulation work.

Part I

Analysis of privacy-preserving systems

Chapter 2

Traffic analysis in anonymous communications

2.1 Introduction

Traditionally, computer security focuses on ensuring the confidentiality, integrity and availability of information. When considering secure communications, these properties are mostly achieved through cryptographic means. Protective measures, however, are often applied only to content, leaving network layer information, such as the identities of the participants in the communication (IP addresses), their location, the amount and timing of data transferred, or the duration of the connection, accessible to possible observers. These, commonly known as traffic data, can be exploited to deduce potentially sensitive private information about the communication.

For instance, in an e-health context messages are generally encrypted to preserve patients' privacy. Yet, the mere fact that a patient is seen communicating with a specialized doctor can reveal highly sensitive information even when the messages themselves cannot be decrypted. Another example is e-voting, where the ability to link voter and ballot not only interferes with the very purpose of the application, but can lead to citizens being subject of coercion fearing that governments or other organizations take reprisals against them.

Confidential communications are not only desirable for personal reasons, but they also play an important role in corporative environments. The browsing habits of a company's employees (e.g. accessing a given patent from a patent database), can be used to infer the company's future lines of investment, thus giving advantage to their competitors. The unstoppable growth of pervasive computing has worsened the problem. It is becoming the norm that high executives carry smart devices (e.g., smart phones, Blackberries, iPods, iPads, etc.) that allow them to be permanently connected to the Internet, thus revealing their location to the service provider. As well as with their browsing habits, studying the movements of these executives can reveal their firm's intentions. For instance, high-level employees from two companies frequently seen together can indicate an imminent acquisition of one company by the other, or a merger of both companies.

Anonymous communications' main goal is the protection of some traffic data, more precisely, they aim at concealing who speak to whom. Guaranteeing anonymity in network communications is harder than just achieving a secure channel. There have been proposals in this field for anonymous email [47] and anonymous Internet browsing [123]. Further research resulted in the deployment of systems like Mixmaster [192] or Mixminion [72] for email, and JAP [28] or Tor [93] for web browsing that attract and increasing number of users.[1] These systems rely on a centralized architecture, but distributed approaches based on peer-to-peer networks have also been proposed [185, 187, 189, 199, 215, 227].

In this part of the thesis we are not overly concerned with the design of anonymity systems, and we refer the reader to [69, 104] for a comprehensive survey of their features. We focus on the analysis and exploitation of the traffic data leaked by the different schemes in order to uncover communication partners, enhance the tracking of messages through the network, etc. Studying how these information leakages affect anonymity is essential in understanding the security offered by these systems. The lessons learned are key to design new systems that incorporate countermeasures to prevent these leakages.

We note that anonymous communications systems are also vulnerable to attacks other than traffic analysis. For instance, Shimshock *et al.* demonstrate in [245] that Minx, an encryption protocol and packet format proposed by Danezis and Laurie [73], has a flaw that allows an adversary to recover the content of the messages. Other attacks that exploit vulnerabilities in the cryptography used by anonymity systems to de-anonymize communications can be found for instance in [62, 63, 218, 280].

2.2 Traffic analysis: basics and applications

Traffic analysis is the process of analyzing intercepted messages in order to extract information from communication patterns. It concentrates on exploiting traffic data (number of messages, frequency, timing, identities of communication partners,

[1] As of February 2011 Tor is used by roughly 250 000 (see `http://metrics.torproject.org/users.html`)

etc.) regardless of the content of the messages. Thus, it can be performed even when messages are encrypted and cannot be decrypted by the adversary. Even though the information obtained is not as profitable as the messages' content, the key quality of traffic analysis is that, compared to cryptanalysis, its deployment is inexpensive and efficient. Usually, not all communications in a system are of interest for the adversary. For instance, in a military context it may be of interest to cut off the enemy's communications, but jamming the full frequency spectrum would require a great deal of power. Instead, traffic analysis can be used to select targets, e.g., it can be used to identify chain-of-command communications. Once targets are selected, the use of more expensive methods such as surveillance, jamming, or destruction, can be optimized instead of wasting resources in attacking all communications: by interrupting the chain of command the enemy operations can be stopped, even if enemy soldiers inside a unit can communicate.

The amount of information that can be derived from traffic data is impressive, and its potential has been recognized in a broad range of areas as noted by Danezis and Clayton [65]. Communication patterns are a source of information about the intentions and actions of the communicators. Some representative examples are:

Who talks to whom. Communicating with a particular person or entity can reveal commercial intentions when two companies start contacting each other, sexual orientation when contacting people from a specific community, or that instructions are being delivered in a military context.

Who talks where. The location where communications take place can reveal medical status of a patient visiting a clinic, political affiliation when a person visits the headquarters of a given party, or the movement or troops in a war scenario.

Who talks how much. Frequent communications can reveal that two people are involved in a relationship, or imminent actions or planning in a commercial or a military context.

Who does not talk. Absence of communications offers as interesting information as its existence. It can reveal the completion of a plan, or a lack of activity.

The military forces were the first in acknowledging the importance of traffic analysis in warfare, as it enables the location of targets and their movements [138]. In their book on wiretapping Diffie and Landau concluded that *"traffic analysis, not cryptanalysis, is the backbone of communications intelligence"* [90]. A cute example of the importance of traffic analysis in the military environment was the deception scheme deployed by the Allied army to mislead the Germans into believing that the invasion would take place at the Pas de Calais, instead of

Normandy. For this purpose, the Allied forces developed a radio net aimed at faking a non-existent army ostensibly directed to Calais.

Even though traffic analysis has military roots, civilians also profit from its power. Social Networks analysis [21, 107, 279] is increasingly gaining importance. Traffic analysis can determine the position of an individual in the social network, which encodes information about his or her status. As a matter of example, Jernigan and Mistree showed that they could infer the sexual orientation of Facebook[2] users by analyzing their friendship links [154]. Another application of traffic analysis is in criminal investigations. For instance, the techniques proposed by Carley*et al.* [45, 97] can be applied to the search of terrorist cell's leaders and efficiently disrupt the chain of command by removing key nodes of the network. The importance of social network analysis to national security is further illustrated by the support given to Visible Technologies by In-Q-Tel, the CIA's investment firm related to access cutting-edge technologies.[3] Visible crawls posts and conversations taking place on web sites such as Flickr, YouTube, Twitter, Amazon[4] or common blogging sites; and analyzes them to determine the importance of authors or conversations.

The connectivity of social graphs (or peer-to-peer graphs) is also key for network security. Traffic analysis exploiting graphs' connectivity can be used for the detection of botnets [198], Sybils [74, 288, 289], or for intrusion detection [172]. Web services can also benefit from traffic analysis. For instance Google's PageRank [213] algorithm that gives a higher rank to pages pointed by a large number of links based on the assumption that they represent hubs, i.e., central nodes, in the network; or Facebook, that improves users' social experience by suggesting them new friends based on their connections in the network.

Traffic analysis is specially relevant for privacy. As we have pointed out, when, where or with whom Alice has contact leaks sensitive information. A trivial approach to protect Alice's privacy could be to decouple her identity from her actions such that they appear as anonymous. However, experience has shown that effectively dissociating users from their communications is a hard task. This is because of the difficulty of eliminating information leaked by traffic data, that can be exploited to reduce the anonymity of the users.

In the rest of this chapter we concentrate on the application of traffic analysis to the de-anonymization of anonymous communications. We note that a deep understanding and knowledge of traffic analysis techniques is essential for anonymous communication systems designers. On the one hand, it allows to evaluate the level of protection achieved by the system under study. On the other hand, it is essential to obtain the probability distributions needed to compute

[2] http://www.facebook.com/

[3] http://www.iqt.org/technology-portfolio/visible_technologies.html

[4] http://www.flickr.com/, http://www.youtube.com/, http://www.twitter.com/, http://www.amazon.com/.

anonymity metrics [52, 81, 82, 103, 238] that can be used to compare the system under study with alternative designs.

2.3 Traffic analysis in anonymous communications

When considering anonymous communications, that intend to hide communication partners, the main goal of traffic analysis is to uncover relationships taking place over an anonymous communications network. In this section we overview the analysis techniques applied to anonymous communications proposed in the literature. We defer the discussion on anonymous communications designed for location privacy to Chapter 6.

We classify the techniques depending on the principle according to which they work. For each of the attacks we also indicate whether it works against high- and/or low-latency anonymous communications systems, and the assumptions on the power the adversary needs to perform the attack. Following the taxonomy proposed by Raymond [224], we consider an adversary as *global* if she has access to the entire communication system (e.g., all communication links), and as *partial* if she only sees part of the network (e.g., a limited number of peers in a peer-to-peer network). We also say that the adversary is *passive* if she only observes the communication and/or controls some of the entities participating in the protocols but does not interfere with the communication. If the adversary can add, delay, alter or remove messages from the system, then we say she is *active*.

2.3.1 Intersection attacks

One of the main building blocks for high-latency anonymous communications is the mix. Introduced by Chaum in 1981 [47], a mix is a router that hides the links between its inputs and its outputs by altering the appearance and timing of the messages. Mixes are suitable to build message-based systems, e.g., email [72, 192], that do not have latency restrictions.

Mixes offer good protection against a global passive adversary assuming that users only send one message through the network [72]. However, if the adversary can observe many messages from one sender to the same set of receivers traversing the network she can perform an *intersection attack* [224] to disclose this sender's communication relationships. In turn, this information can be used to reduce the anonymity of the each of the messages' receiver. This attack relies on the fact that users typically communicate with a reduced number of contacts. Hence, the adversary can find the likely contacts of a target user by intersecting the anonymity sets of the sent messages. Instead of the anonymity set of received or sent messages, intersection attacks can also exploit the fact that different messages use the same route through the network to perform traffic analysis [30].

A type of intersection attack, the Disclosure Attack, was introduced by Kesdogan *et al.* [5, 158] in 2002. The Disclosure Attack reveals the set of friends of a target user, Alice, who communicates through a single threshold mix [47]. This mix follows a very simple mixing process: it collects t input messages, applies a cryptographic transformation to change their appearance, and outputs them in a unique batch to eliminate timing correlations. In spite of the mixing, an adversary observing enough rounds of communications can recover the set of Alice's contacts by intersecting the sets of recipients of batches in which Alice had participated. The downside of this attack is its computation complexity, as it relies on solving an NP-problem. A family of Hitting Set Attacks [160, 161, 176] that looks for unique minimal hitting sets in order to accelerate the search, are able to find the solution more efficiently.

A statistical variant of these attacks was proposed by Danezis [60], known as Statistical Disclosure Attack (SDA), that outputs a very good estimation of a user's sending profile (i.e., the user's preferences when choosing a recipient for her messages. The attack is based on the observation that for a large enough set of observations, by the Law of Large Numbers, the average of the probability distributions describing the recipient anonymity [238] of Alice's messages offers a very good estimation of her sending profile. The SDA was later extended to the analysis of pool mixes [76], to traffic containing replies [70], to recover both the sending and receiving profiles of Alice [181], and to evaluate more complex user models [183].

In 2008, it has been shown that co-inferring users' profiles and assignments of messages between senders and receivers yields better results than doing it independently for each user. We introduced in [270] the Perfect Matching Disclosure Attack (PMDA) and the Normalized Statistical Disclosure Attack (NSDA), explained in detail in Chapter 3. Both attacks improve the results obtained in previous works by considering all senders and receivers simultaneously, accounting for the fact that relationships between sent and received messages must be one-to-one. The PMDA looks for perfect matchings in the underlying graph describing the potential relationships between senders and receivers of messages, while the NSDA considers interdependencies between senders and receivers by normalizing the adjacency matrix representing this graph.

The aforementioned attacks are limited, in the sense that they are only effective under unrealistic assumptions. Early versions of the disclosure attack [5, 60, 70, 76, 158, 160, 161, 176] operate under the assumption that users communicate with a fixed set of contacts and choose the recipient of each message with uniform probability amongst them. Nevertheless, Wilson *et al.* have shown that it is unlikely that in the real world users communicate this way [281]. The PMDA and the NSDA do not present this limitation, but they can only handle simple systems. We propose in [79] the use of advanced Bayesian sampling techniques, further explained in Chapter 4 to handle the simultaneous inference of profiles and assignments in arbitrarily complex anonymity systems. The Bayesian approach has the additional advantage that it provides the analyst with reliable error estimates, hence giving an estimation of the confidence she can put on the results of the attack.

Intersection attacks are also effective in the partial adversarial model. For instance, the predecessor attack, introduced by Wright *et al.* [282, 283], allows an adversary who controls a fraction of the network to keep track of a persistent connection (e.g., VoIP) over multiple path rebuilds in low-latency communications systems such as Onion Routing [93, 124] or Crowds [226]. Then, the identity of the source and destination of the communication can be inferred from their frequent appearances on the observations. The effectiveness of this attack was corroborated by Bauer *et al.* in [22]. Further they show that the attack is even more dangerous, as it can be successfully mounted by a low-budget attacker.

2.3.2 Traffic confirmation attacks

The attacks described in the previous section are effective when users communicate repeatedly with their contacts. Nevertheless, a passive adversary can also trace communications when they are sporadic. An adversary with access to traffic flows can exploit the traffic shape to match incoming and outgoing streams of packets in what is called a *traffic confirmation attack* [258]. These attacks can be applied to both high-latency and low-latency anonymous communication systems, and do not require the adversary to observe all communications (i.e., the adversary can be global or partial).

Different features are useful when it comes to correlating streams. The simplest technique consists in counting the packets of each connection arriving into the network during a time interval, and the number of packets of each connection leaving the network, and use this information to assign inputs to outputs [241, 242]. In the same paper, Serjantov and Sewell also propose to use the pattern resulting from a connection start – an increase in the traffic from a incoming link to an outgoing link – to trace streams. A similar attack using the connection end pattern was introduced by Kesdogan *et al.* [159] in order to identify the origin and destination of messages relayed through Stop-and-Go mixes, which delay messages independently to hide correspondences between inputs and outputs.

Danezis also studied Stop-and-Go mixes in [61]. He presents a more general framework for the analysis of these mixes that is also applicable to other mixing strategies that individually delay messages or use minimum delays to allow for real-time communications [39, 93, 112, 227]. The core idea of Danezis' attack is to model the input stream and the delay processes as signals, and use signal detection techniques to identify correspondences. Signal processing techniques were also used by Levine *et al.* to re-identify flows by leveraging the differences in their interpacket delays [173].

Murdoch and Zieliński demonstrate in [196] that in order to use traffic confirmation the adversary does not need access to the whole flow of data. It suffices if the adversary can sample 1 in 2 000 packets on each side of the communication, e.g., an attacker placed at the Internet exchanges. The attack uses simplified Bayesian statistics and operates under the assumption that traffic is Poisson distributed.

Flow correlation is also useful to identify hidden services in the Tor network [93]. These services are hidden in that the client does not know the exact location of the service in the network. In order to communicate with the server, instead of directly connecting through Tor, the client opens an anonymous connection to a rendezvous point that in turn anonymously communicates with the server. Øverlier and Syverson [212] demonstrate that an attacker controlling at least one Tor node can discover the location of a Hidden Server. The attack exploits Tor's random selection of nodes to build connections: the adversary connects to the hidden service repeatedly until the malicious node is the neighbor of the hidden server in the path. At this point simple correlation [61, 173, 196, 242] confirms whether the attack is successful. The authors note that this attack could also be used by a malicious non-hidden server to de-anonymize users, not only in Tor but in other anonymity networks as Freedom [39] or Crowds [226].

2.3.3 Web fingerprinting attacks

One of the simplest attacks a partial attacker can deploy on a low-latency anonymity system is called the web site fingerprinting attack [139, 141, 174]. The attack consists of two phases. First a training phase in which the adversary creates fingerprints for a number of sites (popular or interesting sites) and stores them. This fingerprints represent the web page in terms of packet sizes, timings, counts, etc. The second is a testing phase in which the attacker monitors the encrypted traffic of the user and tries to correlate its shape with the "templates" in her database. For this purpose the attacker only needs to have access to the victim's traffic, and to have knowledge of the identity of the victim (e.g., the source IP address of the traffic). Panchenko and Pimenidis point out that local network administrators, Internet Service Providers, or secret agencies are in this position [214].

This attack can also be performed by a global passive adversary, who of course can observe the victim's traffic. Nevertheless, if the adversary can observe all inputs and outputs of the system she can make use of a confirmation attack (such as the ones explained in previous section), avoiding the need for a training phase.

2.3.4 Routing constraints-based attacks

Anonymity systems impose route constraints on the paths chosen by users. For instance, given the underlying anonymity protocol some attributes of paths are fixed (e.g., in Tor paths are composed by three distinct relays, and the first relay must be of a special type [93]); or given the node discovery algorithm users have a partial view of the network, hence their paths can only be formed by a subset of all the nodes (e.g., Tarzan [112], Salsa [199], or NISAN [215]).

The fact that routing constraints significantly affect the anonymity provided by a system with respect to a global passive adversary was first observed by Serjantov and Danezis [238]. Serjantov noted later on that the problem became intractable when the number of constraints grows [237]. We proposed the use of Bayesian inference sampling techniques [266] to reduce the computational load, and efficiently analyze complex systems. These techniques are explained in detail in Chapter 5.

Danezis and Clayton observe that when users have a partial view of the network (i.e., they only know a subset of nodes) the adversary can use this information to *fingerprint* routes that can only belong to certain users [64]. Later, Danezis and Syverson showed that the adversary can improve this result by *bridging* [77] honest routers: considering the path initiator's knowledge, or ignorance, about subsequent nodes in a route (e.g., there is only one exit node known by the initiator, or there is only one exit node that is not known by the rest of the senders in the system).

The adversary can also try to bias the choice of nodes in her advantage in order to increase the likelihood of controlling both ends of a target's path, and thus be in the position to launch a confirmation attack. Murdoch and Watson demonstrated that an adversary with access to a botnet can easily gain control of a large fraction of the circuits in the Tor network [195]. Further, Borisov *et al.* [38] showed how an adversary with capacity to perform denial of service can lower anonymity using the retransmission of messages to increase the opportunities for attack, thus requiring even less resources.

Peer-to-peer anonymous networks [185,187,189,199,215,227] suffer from the same problem. In these networks nodes must perform lookups in order to find peers to relay their communications. Malicious nodes can misdirect these lookups to ensure that colluding relays are chosen to form a path and violate the anonymity of the system. To avoid these attacks peer-to-peer anonymous communication designs incorporate defenses aimed at enabling the secure lookup of random relays. These defenses, mostly based on redundant checks, allow an adversary controlling a small fraction of the network to observe many non-anonymous lookup messages effectively upgrading him to an almost-global adversary. There is a line of research dedicated to find such flaws in anonymous lookup mechanisms [188,236,260,265, 275], although to the best of our knowledge currently no secure solution has been found.

2.3.5 Watermarking attacks

Another powerful family of attacks that only need partial monitoring of the network to trace flows in low-latency anonymous communications systems are watermarking attacks. Here the adversary alters some characteristic of a target flow (normally packet timings) in a known fashion, hence introducing a "watermark." The adversary then searches for this watermark in other flows. Flows containing the watermark are considered to be the same. The most popular are interval-based watermarks [165, 221, 276–278, 290], which modify the inter-packet delays to mark the flows.

2.3.6 Clogging attacks

Clogging attacks are similar to watermarking attacks, in the sense that the adversary manipulates the shape of the traffic to be able to re-identify flows of packets. Clogging permits a partial adversary to find the route followed by a target stream of messages in low-latency communication systems. In the simplest version of this attack [15] an adversary sequentially clogs nodes suspected to be in the victim's path, and looks for the corresponding decrease in bandwidth at the client's connection in order to identify which of the suspects actually belongs to the path.

A low-cost version of this attack applied to the Tor network was introduced by Murdoch and Danezis [194], and extended by McLachlan and Hopper to peer-to-peer networks [184]. In this attack, users can be de-anonymized by a malicious server that colludes with a malicious Tor node. The malicious node acts as a probe that clogs and unclogs (i.e., it mimics a pulse) the other Tor routers. When the correct node is probed, the latency observed at the targeted output has a high correlation with the pulses introduced by the probe, signaling the members of the route followed by the stream. This attack was extended to use circuit latency measurements in order to infer information about the position in the network of the client by Hopper *et al.* [145].

2.3.7 Blending attacks

A global active adversary with the capability to delay messages can deploy the so-called blending attack, also known as n - 1 attack [207, 237, 239] on high-latency mix-based anonymity systems [47, 72, 192]. The attack targets one message, for which it aims to identify the receiver. When the target message enters the mix, the adversary delays every other incoming message. Simultaneously she generates many messages such that the only message unknown to her in the batch is the target message. Hence, when the mix flushes it is trivial to pinpoint the target amongst the outgoing messages.

2.4 Conclusion

In this chapter we have defined what traffic analysis is, and we have reviewed its historical roots. Then, we have given an overview of the relationship between traffic analysis and anonymous communications. We have seen that traffic analysis is a very powerful threat for all types of anonymity systems, regardless of the adversarial model considered in their designs. In the following chapters we dive into a deeper discussion of our contribution to the analysis of anonymous communication systems.

Chapter 3

Perfect matching disclosure attacks

3.1 Introduction

Relay-based anonymous communications were first proposed by David Chaum [47], when he introduced the concept of *mix*, and have since been the subject of considerable research [69,104] and deployment [72,192]. A mix is a relaying router that hides the relation between incoming and outgoing messages. For this purpose it collects messages, transforms them cryptographically, and outputs them in a randomized order. The cryptographic transformation prevents bitwise linkability: the input and output messages appear different to a passive observer such that connections between them based on the bits they contain is not possible. On the other hand, the shuffling of messages prevents linkability based on the timing and order of outgoing messages with respect to the inputs received by the mix. In this thesis, in order to present our traffic analysis techniques, we assume the cryptography is perfect and leaks no information, even though there have been attacks in the literature that demonstrate that this is not always the case [245].

Mixes provide anonymity at the cost of substantial delays in the communication, and therefore they can only be used for applications that tolerate high latencies, such as anonymous email [72, 192] or e-voting protocols [152, 230]. They are however not suitable for interactive applications with low-latency constraints, such as web browsing or instant messaging, for which low-latency techniques have been developed [28,93,123,199]. For further details on the extension and refinement of mix-based protocols we refer the reader to the survey by Danezis *et al.* [69,104].

In parallel with the development of mix networks, techniques to uncover persistent and repeated patterns of communication through them have been proposed. Such attacks were first named "intersection attacks" [224] (see Sect. 2.3.1) since they were based on the idea that when a target user repeatedly communicates with a single friend it is possible to uncover the identity of the latter by intersecting the recipient anonymity sets of this user's messages.

Kesdogan *et al.* [5,158,161] introduced a family of *disclosure* and *hitting set attacks* that generalizes this idea to users with multiple friends. After the adversary observes a number of messages going through the system, the outcome of these attacks is the set of friends of each sender being uncovered. Statistical variants of these attacks were also developed, known as *statistical disclosure attacks* [60], and applied to pool mixes [76], traffic containing replies [70], to recover both the sending and receiving profiles of Alice [181], and evaluated against complex models [183]. This family of attacks operates regardless of the internal details of the network, thus they represent a fundamental limit on the level of protection that an anonymity network can provide against traffic analysis, and it is likely that they can only be avoided in a very inefficient manner by introducing dummy traffic in the network [29].

Although these attacks are considered as a key reference in the evaluation of new designs for anonymous communications, their effectiveness strongly relies on unrealistic assumptions such as "users pick their communication partners uniformly at random." These assumptions simplify the calculation of anonymity, and hence aid our understanding and intuition of how the traffic data leakage can be exploited by an adversary. However, even though they make the model optimal to illustrate the principles behind the attacks, the deviations from real world usage jeopardize the validity of the results obtained with respect to real implementations of anonymous communication networks.

Human behavior is hard to model and predict, and even the most sophisticated adversary with access to vast amounts of information can only at best approximate user behavioral profiles. Furthermore, due to the lack of available real-world data in the academic community, little is known about how user sending profiles might actually look like, or how they evolve in time. The first original contribution presented in this chapter is a non-restrictive user behavior model in which users have an arbitrary number of friends amongst which the recipient of every message is chosen with arbitrary probability. The flexibility of this model allows us to evaluate systems in more realistic scenarios than prior work [5,60,70,158,161,183].

Disclosure attacks were originally designed to obtain users' communication patterns. We call a user's communication pattern her *profile*, thus the goal of the adversary is to perform *profiling.* The profiles include the set of contacts with which users communicate, together with the probability distribution that describes the users' preferences amongst them. The sending profiles derived in the attack can be used to improve the capability of the adversary to individually trace each of the messages sent to the network [60, 70]. We denote the process of uncovering the sender of a received message as *de-anonymization* of that message.

Our second contribution is to show that the effectiveness of the Statistical Disclosure Attack (SDA) [60], considered as the most efficient of the disclosure attacks' family [5, 60, 70, 76, 161], is strongly dependent on the users' behavior. We empirically compare the performance of the SDA when tracing messages in presence of different user behavior models. We show how the SDA's performance worsens as the user behavior progressively differs from the model considered by Danezis when designing the attack [60].

The third contribution in this chapter is a more advanced analysis technique, the Perfect Matching Disclosure Attack (PMDA), that obtains good success rates when de-anonymizing messages regardless of the user model considered. The key idea that makes our attack more efficient is that it considers all users in a round at once, instead of focusing on individual users. We compare the SDA and the PMDA through simulation and show that our method is more accurate in de-anonymizing messages.

The PMDA, although more effective than the SDA, requires expensive computations that may make the attack infeasible for scenarios with a large number of users. For these cases we introduce the Normalized Statistical Disclosure Attack (NSDA). The NSDA trades off between precision and speed, yielding results nearly as good as the PMDA with a running time slightly higher than the original SDA. If precision is needed, or if the system under analysis is more complex than the threshold mix considered in this chapter, we refer the reader to the traffic analysis techniques described in Chapters 4 and 5, previously published in [79, 266]. These techniques are based on the same basic principle as the PMDA, i.e., consider all information available to the adversary at once, but they use more powerful and flexible mathematical tools.

Finally, we propose an enhanced profiling methodology which uses the outcome of the attacks (i.e., the result of the de-anonymization in each of the observed rounds) as input for a new profiling step, allowing the adversary to derive more accurate estimations of users' profiles. We show how this technique can be combined with any of the attacks considered, improving the quality of the profiles obtained in all cases.

The results presented in this chapter have been extracted from our original work *Perfect Matching Disclosure Attacks* published at the *8th Privacy Enhancing Technologies Symposium* [270]. Further, the findings presented in this chapter served as inspiration for later results [79, 89, 121, 266] as pointed out along the chapter.

Chapter outline

This rest of this chapter is organized as follows. We explain the system model and the user behavioral models in Sect. 3.2. Section 3.3 and Sect. 3.4 describe the original Statistical Disclosure attack, and the Perfect Matching Disclosure attack, respectively. We evaluate both methods in Sect. 3.5. We explain in Sect. 3.6 how to construct enhanced user profiles and present the Normalized Statistical Disclosure Attack in Sect. 3.7. Finally, we discuss some open questions and conclude in Sect. 3.8.

3.2 System model

We consider a system where a set U of N_{user} users send messages to each other through an anonymous communication channel $\mathcal{A}$. In this chapter we consider this channel to function as a threshold mix. This type of mix follows a plain mixing strategy: it collects t input messages, where t denotes the *threshold*, and outputs them all at once after a cryptographic transformation. This transformation ensures that an adversary cannot link inputs and outputs by simple observation. Further, as messages are flushed at the same time, linkability based on the arrival/departure time of messages [61, 76] is prevented.

We define the *sending profile* of a user $x \in U$ as the vector of probabilities Ψ_x of size N_{user}. A given element of this vector expresses the probability that x chooses $y \in U$ as the recipient of one of her messages. As any well-defined probability distribution, the sum of its component adds up to 1, i.e. $\sum_y \Psi_x(y) = 1$ for all x. Lets say that x is Alice, and that y is Bob. Then, $\Psi_{\text{Alice}}(\text{Bob})$ is the probability that Alice chooses Bob as recipient when she sends a message. The distribution as a whole describes Alice's sending behavior with respect to the entire population (including herself). We use the following notion of *friendship*: we say y is a friend of x, if x sends a message to y with non-zero probability (i.e., $\Psi_x(y) > 0$).

As in previous work [70], we model the sending rate of each individual user $x \in U$ as a Poisson distribution with parameter λ_x. In general, we do not expect real users to initiate discussions in a way that can be approximated by a single Poisson process. There will definitely be fluctuations in the communication rate according to the time of day, the week day, the environment of the user and the user herself. Nevertheless, the Poisson's memorylessness property makes it ideal to set up a framework in which we can concentrate on studying the attack's properties regardless of the event generation scheduling.

3.2.1 Adversarial model

We consider a global passive adversary that monitors the system and observes all input messages arriving to the mix (and their respective senders), as well as all output messages leaving the mix (and their recipients), but not the internal operations of the mix. Naturally, the messages are encrypted so the content is hidden. The attacker observes N_{msg} messages sent through the system, divided into ρ disjoint rounds of equal size. We denote Sen_{I_r} the set formed by the senders of the t messages arriving at the mix in round r and Rec_{O_r} the set of the corresponding receivers. We denote the whole set of ρ round observations as the trace $\mathcal{T} = (I_r, O_r), 1 \leq r \leq \rho$.

Although the attacker does not know the correspondence between inputs and outputs, she is able to compute the probability distributions linking every input with all possible outputs and vice versa. Computing this distribution for threshold mixes is straightforward as in each round every incoming message has an equal probability of corresponding to each outgoing message.

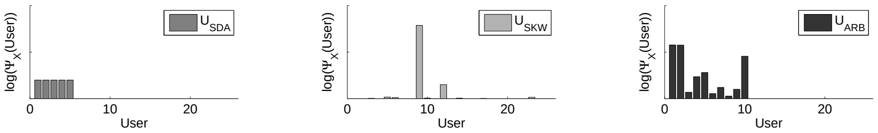
The adversary uses this probability distribution and the information in $\mathcal{T}$ for two purposes. Her first goal is to recover the users' sending profiles, i.e. discover the users' preferences when choosing a recipient for their messages. Secondly, she aims at linking back the inputs and outputs of every round, i.e. uncover who communicated with whom while the system was under observation.

3.2.2 A non-restrictive user behavior model for anonymous communication networks

We consider three types of populations. The first one, U_{SDA}, is a simple and very restrictive user behavior model inspired by the one used by Agrawal and Kesdogan [5], and Danezis [60]. Modifying our assumptions on the number of users' friends and the user preferences amongst them, we construct two additional populations U_{SKW} and U_{ARB} that gradually differ from U_{SDA}. We define the models as follows:

U_{SDA}**:** a single user, Alice, has f randomly selected friends; her sending behavior toward her friends is uniform; Ψ_{Alice} contains f times the value $\frac{1}{f}$ and $N_{\text{user}} - f$ times the value zero; all other user profiles contain N_{user} times $\frac{1}{N_{\text{user}}}$. Figure 3.1(a) depicts this profile.

U_{SKW}**:** every user x has an individual number f_x of friends; the sending probabilities toward the friends are generated such that the resulting profile is skewed [281]. This means that users have a set of contacts where there are one or two very good friends (which they choose as recipients in more



(a) An example of U_{SDA} profile

(b) An example of U_{SKW} profile

(c) An example of U_{ARB} profile

Figure 3.1: Examples of user behavior

than 60% of the cases) and the rest have small probability of being chosen. Figure 3.1(b) depicts this profile.

U_{ARB}: every user x has an individual number f_x; the sending probabilities toward the friends are generated such that users do not have strong preferences about their contacts, still, their distribution is non-uniform. Figure 3.1(c) depicts this profile.

We have developed further variants of these models to focus on different features of user behavior which we use in [89] to evaluate the impact of social network profiling on anonymity.

Comparison with previous models

The original Disclosure Attack and its first sequels [5, 60, 161] consider a model that is almost equivalent to our model with population U_{SDA}. In their model, Alice sends exactly one message in each of the rounds in which she participates. As we demonstrate in [121], the fact that users may send and receive multiple messages per round influences the anonymity provided by a system. Therefore, we remove this limitation in our model and let users send an arbitrary number of messages per round of communication.

Mathewson and Dingledine employ simulations in order to evaluate the effectiveness of statistical disclosure attacks when it comes to recover profiles from traffic data [183]. In their work they introduce a more complex model than the one in the seminal paper [5]. The two main differences with respect to the original model are: i) Alice is allowed to send more than one message in each round in which she participates; and ii) every participant has a set of friends (as opposed to the U_{SDA} model in which all other users' profiles contain N_{user} times $\frac{1}{N_{\text{user}}}$). Nevertheless, their behavior toward them is still uniform, i.e. users send the same volume of messages to all their friends. In some of their experiments Mathewson and Dingledine go a step further and let Alice choose with non-uniform probability amongst her friends, but not the rest of the users, obtaining a model closer to our U_{ARB}.

The Two-Sided Statistical Disclosure Attack [70], another variant of the Disclosure Attack, is tested under U_{SDA} traffic and in a variant where all users have the same number of friends, to which they send with uniform probability. Both models allow Alice to send several messages per round in which she participates.

The main drawback of these models is their narrowness. Our model U_{ARB} aims at covering a more realistic range of scenarios. In particular, utilizing U_{ARB} implies no assumption about the number of users that have friends, the number of friends they have, or the sending behavior toward their friends.

3.3 The Statistical Disclosure Attack: profiling and de-anonymization

3.3.1 Profiling with the Statistical Disclosure Attack

The Statistical Disclosure Attack (SDA), as presented by Danezis in [60], focuses on revealing the *likely* set of friends of a target user, Alice. Alice is the only user in the system who has a limited number of friends (Ψ_{Alice} contains f positions with value $1/f$ corresponding to her f friends), and the rest of the population choose their recipients uniformly amongst all the users ($\Psi_{\text{Sen}_I}(\text{Rec}_O) = \frac{1}{N_{\text{user}}}$ for all $i_r \in I, o_r \in O$, $\text{Sen}_I \neq$ Alice).

In each round r where Alice is sending a message, an attacker deploying the SDA computes the probability distribution Θ of the potential recipients of this message as a combination of the profiles of all the participating senders as follows:

$$\Theta_r = \frac{1}{t}\Psi_{\text{Alice}} + \frac{t-1}{t}\Psi_x,\ x \in \text{Sen}_{I_r} \setminus \{\text{Alice}\}\,. \tag{3.1}$$

For a sufficient number ρ of observed rounds, the law of large numbers allows to estimate Alice's profile from the empirical mean over the observed rounds:

$$\bar{\Theta} = \frac{1}{t}\sum_{r=1}^{\rho}\Theta_r \approx \frac{\Psi_{\text{Alice}} + (t-1)\Psi_x}{t} \Rightarrow \tilde{\Psi}_{\text{Alice}} \approx t\frac{\sum_{r=1}^{\rho}\Theta_r}{t} - (t-1)\Psi_x\,.$$

Using the round observations contained in $\mathcal{T}$ as input to this method, the attacker estimates the profiles of all the users in the system. We denote the estimated profile of user x obtained in this phase $\tilde{\Psi}_{x,SDA}$, for each user x in the population, and we denote the whole set of these profiles as $\tilde{\Psi}_{SDA}$.

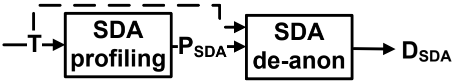


Figure 3.2: De-anonymization with the Statistical Disclosure Attack

3.3.2 De-anonymization with the Statistical Disclosure Attack

As suggested in [60, 70], the estimated profile can be used to rank the potential receivers of a message from Alice according to the likelihood that Alice would send to them. The most likely receiver Rec_k of her message in a round r can thus be easily identified as

$$\mathrm{Rec}_k = \mathrm{argmax}_{\mathrm{Rec}_k}\ \tilde{\Psi}_{\mathrm{Alice},SDA}(\mathrm{Rec}_k),\ \mathrm{Rec}_k \in O_r\,. \tag{3.2}$$

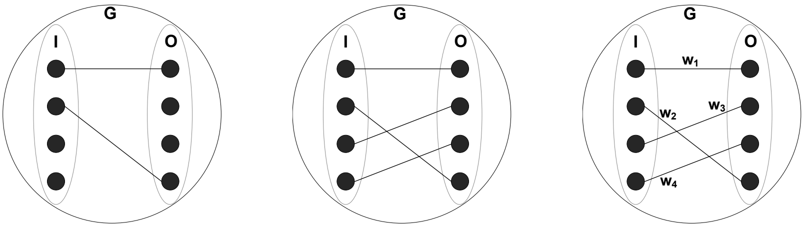
When de-anonymizing the receivers of several messages in one round, the most obvious, though naïve approach is to repeat this procedure for each individual message. Figure 3.2 depicts the entire de-anonymization process, where the box marked as SDA profiling represents the profiling step described in the previous section, and the output D_{SDA} is the de-anonymization result of the attack.

3.4 The Perfect Matching Disclosure Attack

In this section we first recapitulate the required basic notions of graph theory needed to understand the foundations of the Perfect Matching Disclosure Attack (for further reading about graph theory in an anonymity context we refer the interested reader to [148]). Then, we show how a threshold mix can be modeled using bipartite graphs. Finally, we explain how maximum weighted bipartite matchings can be used to efficiently de-anonymize users communicating through this mix if the attack is transformed into a classic optimization problem.

3.4.1 Basic graph theory notions

A graph $G = (N, E)$ consists of a set of nodes N and a set of edges E. Without loss of generality we assume $N \neq \emptyset$. A bipartite graph $G = (I \cup O, E)$ is a graph whose nodes can be divided into two distinct sets I and O such that every edge in E connects one node in I and one node in O. In other words, there exists no edge between nodes from the same set. In this chapter we focus on sets I and O of equal and finite cardinality $t > 1$. A set of edges $M \subseteq E$ is called a matching in the bipartite graph G if no node in G is incident to more than one edge. A *perfect matching* additionally requires that every node is incident to exactly one edge.



(a) A matching

(b) A perfect matching

(c) A weighted perfect matching

Figure 3.3: Bipartite graphs

If each edge $e_l \in E$ is associated with a weight w_l, the graph is denoted a *weighted bipartite graph*. A maximum weighted bipartite matching is defined as a perfect matching for which the sum of the weights w_l associated with the edges in the matching has a maximal value, *i.e.* the perfect matching M maximizes $\sum w_l \; |e_l \in M$. Figure 3.3 illustrates the definitions. In the rest of this work we focus on maximum weighted bipartite matchings and assume completeness of the graph. If the graph is not complete bipartite, *i.e.* edges are missing, one usually inserts the missing edges with an associated weight of zero.

Finding such matchings is often called the *assignment problem*, one of the fundamental combinatorial optimization problems in graph theory. Further, one deals with a *linear* assignment problem when the two following conditions are met: i) the sets of nodes I and O are of equal and finite size and ii) the total weight of the assignment (or matching) is equal to the sum of the weights associated to the edges in the assignment.

3.4.2 The optimization problem in the anonymous communication setup

We represent each of the t messages i_j, $j = 1, \ldots, t$, sent to the mix during one round as a node. These nodes form the set $I = \{i_j\}$, and we label them with their corresponding sender's identities Sen_j. Note that a node does not represent a specific user, but a message sent by a specific user. Therefore, two messages from one sender are represented by two different nodes with the same label. Equivalently, the t messages received during one round form the set O where each node o_k is labeled with the receiver's identity Rec_k, $k = 1, \ldots, t$.

We model the relationship between an incoming message i_j and an outgoing message o_k with an edge e_{jk} connecting these two nodes. This edge implies that these two messages are the same (i.e., $i_j = o_k$), therefore exhibiting the link

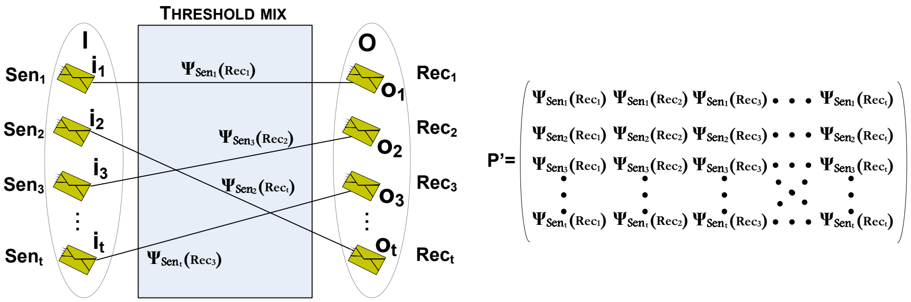

THRESHOLD MIX
I
O
Sen1
Sen2
Sen3
Sent
i1
i2
i3
it
o1
o2
o3
ot
Rec1
Rec2
Rec3
Rect
ΨSen1(Rec1)
ΨSen3(Rec2)
ΨSen2(Rect)
ΨSent(Rec3)
P'=
ΨSen1(Rec1) ΨSen1(Rec2) ΨSen1(Rec3) • • • ΨSen1(Rect)
ΨSen2(Rec1) ΨSen2(Rec2) ΨSen2(Rec3) • • • ΨSen2(Rect)
ΨSen3(Rec1) ΨSen3(Rec2) ΨSen3(Rec3) • • • ΨSen3(Rect)
ΨSent(Rec1) ΨSent(Rec2) ΨSent(Rec3) • • • ΨSent(Rect)

Figure 3.4: Mapping of the optimization problem to the threshold mix environment

between sender and receiver (i.e., Sen_j chose Rec_k as receiver in this round). We assign a weight w_{jk} to these edges, representing the probability that Sen_j actually chooses Rec_k as recipient. These weights are derived from the users' profiles Ψ_x, which can be known to the adversary, or estimated from observations of mixing rounds as we discuss in Section 3.3.1. We recall that Ψ_x describes the sending behavior of user x toward the entire population but, for a given round, only those elements of Ψ_x associated with the receivers in the round are of interest. Therefore we construct the $t \times t$ matrix P', and assign the weights w_{jk} as follows:

$$\begin{aligned} P'(i_j, o_k) &= \Psi_{\text{Sen}_j}(\text{Rec}_k)\,, i_j \in I, \text{Rec}_k \in O; \\ w_{jk} &= P'(i_j, o_k)\,, i_j \in I, \text{Rec}_k \in O. \end{aligned}$$

The nodes $I \cup O$ together with the edges $E = \{e_{jk}\}$ form the complete bipartite graph $G = (I \cup O, E)$. We note that if a different anonymous channel was in place ruling out some sender-receiver combinations, or if the user profiles exclude certain individuals as possible communication partners, the relation would be represented by an edge of weight zero. We illustrate this mapping in Fig. 3.4.

The goal of the adversary is to recover the users' profiles as well as de-anonymize each message that has traveled through the mix. In a nutshell, our idea is to take advantage of the fact that all messages sent and received during one round have to be linked in pairs (every input is linked to one, and only one, output). Further, we are looking for the most likely assignment of inputs and outputs, i.e. the assignment that maximizes the joint probability of the links. Thus, the optimization problem we are facing is to find the maximum weight matching M in G given the sets I and O, and the weights in P'. We denote the space of all perfect matchings on the graph G by $\mathcal{M}$ and require that an eligible set of edges belongs to this space, *i.e.* it must be a perfect matching $M \in \mathcal{M}$.

Applying Bayes theorem, the conditional a posteriori probability $\Pr[M|I,O]$ can be computed as

$$\Pr[M|I,O] = \frac{\Pr[I,O|M] \cdot \Pr[M]}{\Pr[I,O]} \,.$$

Given an assignment M, the sets of nodes I and O are implicitly fixed and thus $\Pr[I,O|M] = 1$. It follows that $\Pr[M|I,O] = \Pr[M]/\Pr[I,O]$. Since the sets I and O are given in the condition, $\Pr[I,O]$ is a constant term and independent of the considered assignment M. Therefore, the assignment M maximizing $\Pr[M|I,O]$ is the one that maximizes $\Pr[M]$.

An assignment M is a perfect matching on G, thus $\Pr[M]$ is the joint probability of the individual edges $e_{jk} \in M$. Assuming that the edges $e_{jk} \in M$ are independent the joint probability $\Pr[M]$, that we want to maximize, is the product of the individual edge probabilities:

$$\Pr[M] = \prod_{e_{jk} \in M} w_{jk} \,.$$

We consider the adversary observes the system during ρ rounds, constructing the trace $\mathcal{T} = (I_r, O_r), 1 \leq r \leq \rho$. Given a round observation, which consists of multisets of senders Sen_{I_r} and receivers Rec_{O_r}, the probability of each assignment M is $\prod_{e_{jk} \in M} w_{jk}$. The assignment M maximizing $\Pr[M]$ also maximizes $\Pr[M|I_r, O_r]$.

In our model we mandate that all senders choose when to communicate according to a Poisson distribution with the same parameter. Thus, all combinations of senders are equally likely to be observed. Besides, each sender chooses the recipient(s) of her message(s) independently of the choice(s) of all other senders. If a user sends multiple messages, the receivers of these messages are also chosen independently. Therefore, our model easily accommodates the case that a user sends two (or more) messages by considering that these messages have been sent by two (or more) distinct senders with identical profiles that each send one message to independently chosen receivers.

3.4.3 De-anonymizing messages with the Perfect Matching Disclosure Attack

As we have explained, the adversary's goal is to find a maximum weighted bipartite matching on the graph representing her observation of messages entering and

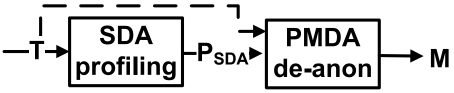


Figure 3.5: De-anonymization with the Perfect Matching Disclosure Attack

leaving the mix. In terms of algorithmic computer science, once the graph and its edges' weights are known, this is an assignment problem with known solution [91, 169].

In order to derive the weights w_{jk} the adversary uses the trace $\mathcal{T}$ to estimate simple user profiles $\tilde{\Psi}_{SDA}$ as described in Sect. 3.3.1, and for each round r, she constructs the $t \times t$ matrix P':

$$P'(i_j, o_k) := \tilde{\Psi}_{\mathrm{Sen}_j, SDA}(\mathrm{Rec}_k)\,, i_j \in I_r, o_k \in O_r\,.$$

With these values, the adversary can compute the joint probability of all t links in an assignment M as

$$\Pr[M] = \prod_{e_{jk} \in M} P'(i_j, o_k) = \prod_{e_{jk} \in M} \Psi_{\mathrm{Sen}_j}(\mathrm{Rec}_k)\,, i_j \in I_r, o_k \in O_r\,.$$

We have shown in Sect. 3.4.2 that the assignment M that maximizes p_{joint} is the adversary's best guess about the correspondences between inputs and outputs. In order to transform the maximization of p_{joint} into a linear assignment problem we replace each element of the matrix $P'(I_r, O_r)$ with its logarithmic value $\log_{10}(P'(I_r, O_r))$ before associating it to the edge e_{jk} linking message i_j to message i_k (see Fig. 3.5):

$$\log_{10}(p_{joint}) = log_{10}(\prod_{e_{jk} \in M} P'(I_r, O_r)) = \sum_{e_{jk} \in M} \log_{10}(P'(I_r, O_r))\,.$$

Having each edge associated with a log-probability, we can use a suitable algorithm to solve linear assignment problems [25, 91, 169] and obtain the most likely sender-receiver combination for all t messages in the round as the perfect matching $M \in \mathcal{M}$.

3.5 Empirical evaluation of de-anonymization techniques

In order to evaluate the performance of the Perfect Matching Disclosure Attack, we deploy it in different scenarios and compare it to the original Statistical Disclosure Attack. Our goal is to study the impact of system parameters on the effectiveness and viability of both attacks.

3.5.1 Experimental settings

Our experiments are carried out on populations U of size $N_{\text{user}} = 1000$ users who send messages through a threshold mix with threshold $t = 100$, ensuring that a considerable fraction of the users participate in each mixing round. Every user $x \in U$ chooses her recipients according to her profile Ψ_x, which depends on the considered user behavior model (see Sect. 3.2.2), and initiates communications with the same frequency λ. We note that the choice of this parameter's value is arbitrary. As long as all users send messages to the network with equal rate, their frequency of appearance as senders does not depend on its precise value. Although real users are expected to send messages with different frequencies, we chose to fix this parameter in order to create an optimal scenario in which to study the effectiveness of the attacks.

We study how the number of rounds observed by the attacker affects the performance of the PMDA and the SDA. Concretely, we concentrate on their effectiveness, efficiency, and scalability.

For the purpose of our studies we have generated 100 000 mixing rounds. An experiment consists of three steps: i) estimating all user profiles $\tilde{\Psi}_{SDA}$ from ρ round observations; ii) de-anonymizing 5000 rounds with the SDA; iii) de-anonymizing the same 5000 rounds with the PMDA.

Tables 3.1 and 3.2 summarize the parameters and their values in the experiments. Note that when only 1000 rounds are available to the adversary, the steps 2 and 3 of our experiments only consider those rounds.

3.5.2 Results

In this section we present the results of our experiments. To measure the effectiveness of the attacks we define two metrics (both metrics are computed over all 5000, respectively 1000, rounds):

Table 3.1: Parameters of the experiments: μ is the average number of messages used to profile one user, γ is the average number of de-anonymization trials per user

ρ Param	$1k$	$5k$	$10k$	$25k$	$50k$	$100k$
N_{user}	1000					
t	100					
μ	100	500	1000	2500	5000	10000
γ	100	500	500	500	500	500

Table 3.2: Parameters of the experiments: User behavior

Population	$\sharp$ friends f	Profile
U_{SDA}	$\{5, 25, 50\}$	Uniform
U_{ARB}	random $[5, 50]$	Non-uniform

Individual success rate: expresses the accuracy of the attack when de-anonymizing the receiver of a message from a particular sender, *i.e.* successfully linking a specific sender to a receiver. It is computed by counting the number of messages sent by each user that have been correctly de-anonymized by the attack. The success rate per sender is then computed by dividing this number by the number of messages sent by each user.

Round success rate: the percentage of links correctly de-anonymized per round. We calculate it as the average number of sender-receiver pairs successfully identified in each round.

We consider that a message has been de-anonymized correctly if and only if the attack has identified the correct receiver of that message.

Population U_{SDA}

We test both attacks in three U_{SDA} populations where Alice has f friends. We look at the influence of the number ρ of rounds used in the profiling step on the success rates of the attacks.

Figure 3.6 illustrates the individual success rates. All users except Alice send uniformly to the entire population; therefore, the attack cannot make inferences about these users' preferences, and the results refer only to Alice's messages and the individual success rate corresponding to these $\gamma = 500$, respectively $\gamma = 100$, messages (see Table 3.1). This limits the number of messages used when computing the graph resulting in fluctuations of the PMDA success rate. We stress that these fluctuations have no statistical significance.

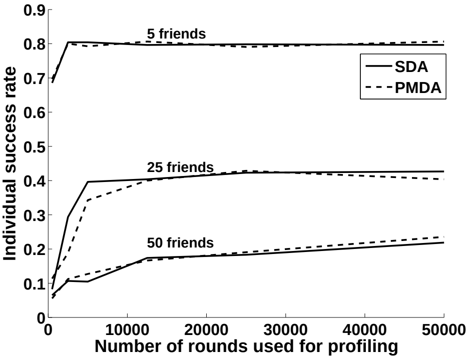


Figure 3.6: Individual success rate in a U_{SDA} population

We can see that both attacks score similarly. On the one hand this is because the rest of senders in the round provide no information. Since their profile is uniform, they give no hints about who Alice is *not* sending to. On the other hand, Alice chooses uniformly amongst her friends. Therefore, if two or more of her friends appear in the set Rec_{O_r}, the best our algorithm can do is to choose randomly amongst them. This last problem also affects the SDA's effectiveness. One can observe in the graph that, the smaller the number of friends (thus the smaller the probability that this case arises) the higher the success rate of both attacks.

As expected, increasing the number ρ of profiling rounds increases the likelihood of successful attacks. It is remarkable, however, that this rate does not increase constantly. When the number of Alice's friends is small ($f = 5$), not much improvement is achieved by increasing the number of profiling rounds above 10 000. Nevertheless, having more rounds helps the attacker when the number of friends increases, as more rounds are needed to observe Alice sending messages to all of her friends.

Populations U_{SKW} and U_{ARB}

Contrary to the U_{SDA} case, where the SDA and the PMDA performed similarly, the PMDA achieves higher de-anonymization success rates when applied to a more general scenario. Figure 3.7 shows the percentage of users participating in the communication for which the attacks obtain a certain individual success rate in both U_{SKW} and U_{ARB} scenarios. We represent different values for the number ρ of rounds used for profiling with different line styles.

We see that the PMDA outperforms the SDA in both experiments, but there is a significant difference between them. With respect to the U_{SKW} case (on the left

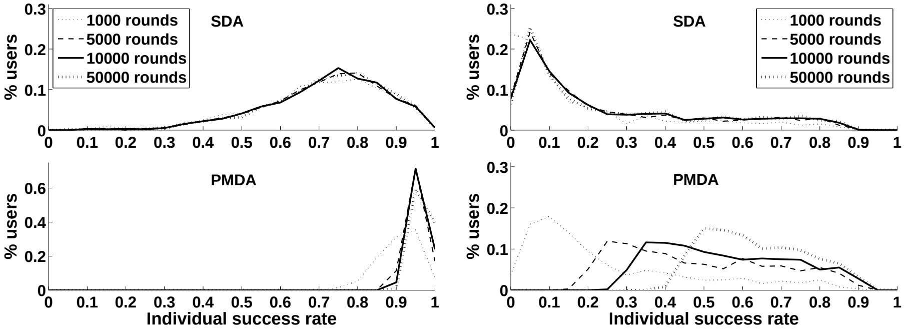


Figure 3.7: Individual success rate attacking U_{SKW} (left) and U_{ARB} (right) populations

side of the figure) and $\rho = 10\,000$ one can observe that the SDA achieves an average individual success rate of 71.5% while the PMDA scores an average individual success rate of 96.04% and de-anonymizes more than 90% of the messages correctly for 99.6% of the users. With respect to the U_{ARB} case (right side of the figure) and $\rho = 10\,000$, the SDA achieves an average individual success rate of 26% while the PMDA scores 55.35%.

Figure 3.8 presents the round success rates of the SDA and the PMDA. Like in the individual success rate, our attack outperforms the SDA. In the U_{SKW} case (left), the SDA has a high rate (71.5% in average) of round de-anonymization, independently of the number of rounds observed. However, the PMDA improves this result de-anonymizing in average 96.05% of the messages in each round when 10 000 rounds have been used for profiling and correctly de-anonymizes the full set of links in 17.22% of the cases. The success of both attacks diminishes when users' sending patterns are uniform toward their friends (case U_{ARB}, right). For the same number of $\rho = 10\,000$ observed rounds the SDA achieves an average round success rate of 25.6% and the PMDA 55.3%.

It is important to note the influence of the number of rounds observed by the attacker on the success rates of the attacks. Increasing the number of observations makes both attacks more accurate. However, the actual improvement notably depends on the type of population attacked. When the attacks are carried out in a U_{SKW} scenario, the users' profiles have a low entropy, thus the strong friends are identified in the first rounds and no additional information is extracted from further round observations. Moreover, the type of attack itself also influences the result. Analyzing a higher number of rounds provides more information, a fact exploited by the PMDA. On the contrary, the SDA's simple decision algorithm takes little advantage of this extra information and we see that almost no improvement is achieved by observing more than 5000 rounds.

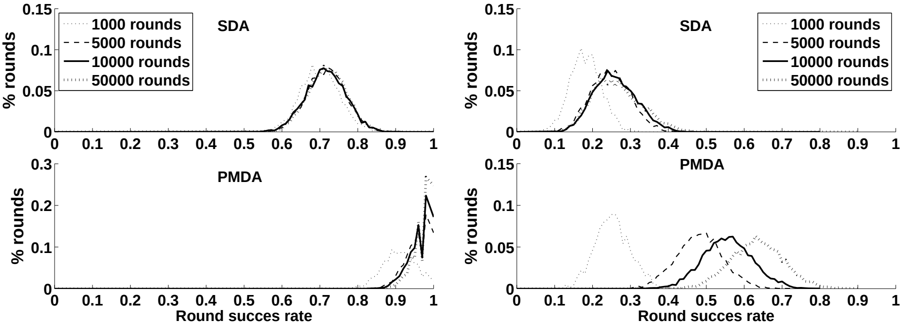


Figure 3.8: Round success rate attacking U_{SKW} (left) and U_{ARB} (right) populations

Table 3.3: Timings of the attacks: estimation of profiles from 50 000 rounds and de-anonymization of 5 000 rounds

Attack	$t = 100$		$t = 500$	$t = 1\,000$
	Time	Success rate, mean (min)	Time	Time
SDA profiling	3.08m	-	38.33m	66.16m
SDA de-anon	10m	25.6% (0.00%)	3.48h	12.91h
PMDA de-anon	10.2m	62.9% (38.8%)	12.9h	4.69days
NSDA de-anon	13.33m	60.2% (33.5%)	4.28h	15.3h

3.5.3 Scalability of the attacks

We evaluate the efficiency of both attacks in terms of time. We implemented both attacks in Matlab, version 7.6.0.324 (R2008a) without any optimizations, using the Hungarian algorithm [169] to solve the linear assignment problem as part of the PMDA.[1] We show in Table 3.3 the time it takes to de-anonymize messages with the SDA (see Fig. 3.2), and with the PMDA (see Fig. 3.5) in U_{ARB} scenarios with mix thresholds $t = 100$, $t = 500$ and $t = 1\,000$, respectively. In all cases the profiles $\tilde{\Psi}_{SDA}$ have been derived from $\rho = 50\,000$ rounds and have been used to de-anonymize 5000 rounds (*i.e.*, find the recipients for all $i_j \in I_r$, $1 \leq j \leq t$, $1 \leq r \leq 5\,000$). We executed our code on a machine with a processor running at 2.8 GHz and 512 KB cache for scenarios with threshold 100 and 500; and on a machine with a processor running at 2.2 GHz and 1 MB cache when the threshold was 1000. The table includes the success rates for $t = 100$ to illustrate the trade-off between accuracy and speed.

The PMDA de-anonymization is slower than the SDA de-anonymization and the

[1] As the Hungarian algorithm aims at minimizing the sum of the edge weights we substitute $P'(\cdot,\cdot) = -P'(\cdot,\cdot)$ before attacking each round.

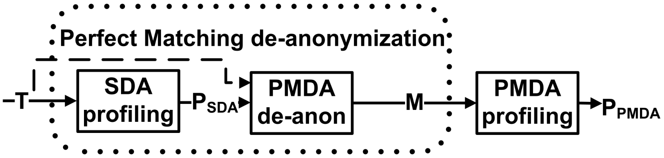


Figure 3.9: Obtaining enhanced profiles with the Perfect Matching Disclosure Attack

difference grows as the size of the threshold, and thus the underlying bipartite graph, increases. Nevertheless, it yields higher success rates. In Sect. 3.7 we propose the Normalized Statistical Disclosure Attack (NSDA), that combines accuracy and speed. Table 3.3 includes the success rate and timings for the operations shown in Fig. 3.12 inside the dotted line. Note that all of the attacks' running times would substantially benefit from optimized implementations. In particular, the PMDA is inherently suited for parallelization.

3.6 Enhanced profiling with the Perfect Matching Disclosure Attack

So far we have focused on the de-anonymization capability of the attacks. In this section, we show how the derived maximum weighted bipartite matchings M_j can be used to better estimate user profiles.

When estimating Alice's profile the SDA considers every receiver in a round as equally likely to be the recipient of a message (see Eq. 3.1). This is because a priori the adversary has no information as to whom are Alice's friends. However, once the PMDA has been performed, the adversary has better knowledge of Alice's preferences. Thus, she can build better estimation of Ψ_{Alice} by considering the receiver(s) indicated by the matching M_j as the most likely, instead of considering all possible receivers of her message(s) in a round r as equally likely. This can be achieved by assigning z to the receiver assigned to Alice's message(s) by M_j and $(1-z)/(t-1)$ to the rest of the elements in Rec_{O_r}. We denote this step as "PMDA profiling" (see Fig. 3.9).

The choice of the weight z, which expresses the confidence on the accuracy of the perfect matchings M_j, is not that crucial. We tested several values for this parameter and observed that its influence on the profile estimation is minor. The only restriction is that the weight z must be strictly greater than $(1-z)/(t-1)$. Choosing $z = (1-z)/(t-1)$ turns the second profiling step useless as this setting is equivalent to the original SDA, and choosing $z < (1-z)/(t-1)$ effectively hides the actual users' relationships. In our experiments we arbitrarily chose $z = 0.5$.

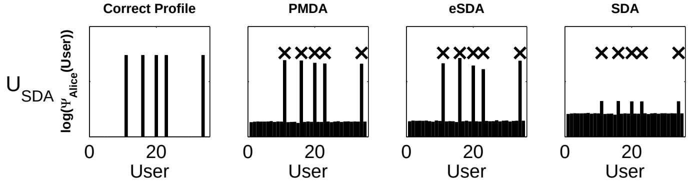


Figure 3.10: Alice's profile and estimations (logscale) for U_{SDA}, $\rho = 100\,000$. From left to right: Ψ_{Alice}, $\tilde{\Psi}_{\text{Alice},PMDA}$, $\tilde{\Psi}_{\text{Alice},eSDA}$, and $\tilde{\Psi}_{\text{Alice},SDA}$

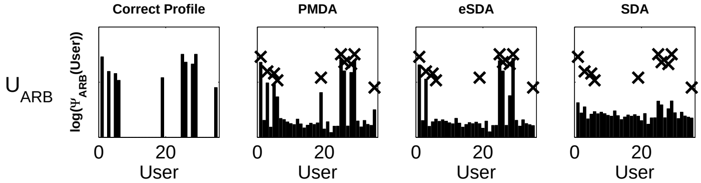


Figure 3.11: Alice's profile and estimations (logscale) for U_{ARB}, $\rho = 100\,000$. From left to right: Ψ_{Alice}, $\tilde{\Psi}_{\text{Alice},PMDA}$, $\tilde{\Psi}_{\text{Alice},eSDA}$, and $\tilde{\Psi}_{\text{Alice},SDA}$

The same procedure can be applied to the decision D_j of the de-anonymization phase of the SDA, yielding a more accurate profile than the one estimated by the original SDA and denoted by $\tilde{\Psi}_{\text{Alice},eSDA}$.

For a U_{SDA} scenario where Alice has five friends ($f = 5$), Fig. 3.10 shows the profile Ψ_{Alice} we initially generated for Alice, her profile after the PMDA's profiling step, the approximation of her profile derived with the enhanced SDA, and her profile estimated using the original SDA. Figure 3.11 shows the corresponding set of profiles for a U_{ARB} scenario.

We observe in both cases that the profile estimation $\tilde{\Psi}_{\text{Alice},eSDA}$ is more precise than $\tilde{\Psi}_{\text{Alice},SDA}$ but not as good as $\tilde{\Psi}_{\text{Alice},PMDA}$.

In the U_{SDA} scenario, all three estimations allow the adversary to easily identify the set of Alice's friends, even if the exact number k of friends is unknown. However, the enhanced methods increase the contrast between friends and non-friends. In the U_{ARB} scenario, $\tilde{\Psi}_{\text{Alice},SDA}$ does not allow to identify friends, and even worse, there exist non-friends of Alice that have higher probability than some of her friends. $\tilde{\Psi}_{\text{Alice},eSDA}$ improves the estimation and allows to identify Alice's best friends (those with high probability in Ψ_{Alice}), but it fails to show more unlikely receivers as for example user 19. The profile $\tilde{\Psi}_{\text{Alice},PMDA}$, on the other hand is a better estimation where all of Alice's friends have higher probabilities than her non-friends.

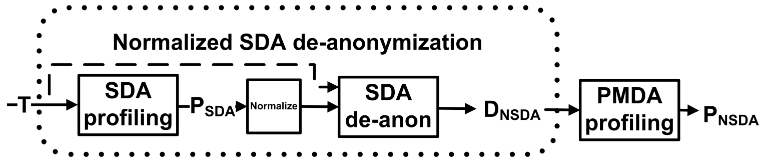


Figure 3.12: Normalized Statistical Disclosure Attack

Note that the same ρ round observations used to construct the simple profile $\tilde{\Psi}_{\text{Alice},SDA}$ are reused to estimate the enhanced profile $\tilde{\Psi}_{\text{Alice},PMDA}$. This straightforward method for reusing information is far from optimal. For instance, if initially the PMDA is misled and outputs a matching M_j signaling erroneous correspondences between senders and receivers, this error will be accentuated when the information is reused. In order to avoid this effect, we propose to use advanced machine learning techniques when information is to be reused [79,266]. A summary of this methodology can be found in Chapters 4 and 5.

3.7 The Normalized Statistical Disclosure Attack

In this section we present a variant of the Perfect Matching Disclosure Attack. The Normalized Statistical Disclosure Attack (NSDA) offers an alternative that trades precision for computational load.

The NSDA, illustrated in Fig. 3.12, has a similar structure as the SDA but it additionally constructs the matrix P' as in the PMDA and it includes a matrix normalization step. The normalization step consists on the transformation of P' into a doubly stochastic transition matrix that, by definition, has the property that each row and each column sums up to one. For this transformation we rely on the method proposed by Sinkhorn in 1964 [250]. He showed that an arbitrary positive $\alpha \times \alpha$ matrix (i.e., each element is greater than zero) can be transformed into a doubly stochastic matrix by iteratively normalizing the rows and the columns of the matrix, and that this iteration converges and has a unique solution. This process is known as iterative proportional fitting.

An element of the normalized transition matrix P' represents the probability of a link between input messages (row) and output messages (column). This ensures that each sent message is received (all rows sum up to 1) and each received message was sent (all columns sum up to 1). The receiver of a given message i_j is chosen as the one who maximizes the individual link probability $P'(i_j, \cdot)$.

The iterative proportional fitting spreads the information contained in each element of P' over the entire matrix, causing two main effects. The first effect is best explained in a noise-free toy example in which the per sender maximum likelihood decision approach of the SDA achieves 66.66% success rate. The matrix P' before and after normalization is the following:

$$P' = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 1 & 0 & 0 \\ 0 & 0.5 & 0.5 \end{pmatrix} \overset{\text{normalize}}{\longrightarrow} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The normalization process over the matrix P' implicitly takes interdependencies between the matrix elements in different rows and columns into account and eliminates impossible combinations. In the toy-example, the certainty $P'(2,1) = 1$ implies $P'(1,1) = 0$. Hence, $P'(1,2)$ becomes 1 to fulfill the doubly-stochastic requirement in the first row. This implies that $P'(3,2)$ becomes also 0 and hence $P'(3,3) = 1$. Therefore, a per sender maximum likelihood decision approach based on the normalized matrix takes more information into account and leads to the only correct assignment with success rate one.

To explain the second effect, we use a noisy version of the same initial matrix P' that contains Gaussian noise with standard deviation 0.1. In this case the per sender maximum likelihood decision of the SDA based on the initial P' leads to the correct assignment for the senders 1 and 2 but to a wrong assignment for sender 3.

$$P' = \begin{pmatrix} 0.4006 & 0.4208 & 0.1786 \\ 0.7810 & 0.1432 & 0.0757 \\ 0.0997 & 0.4580 & 0.4424 \end{pmatrix} \overset{\text{normalize}}{\longrightarrow} \begin{pmatrix} 0.2776 & 0.4369 & 0.2856 \\ 0.6673 & 0.1834 & 0.1494 \\ 0.0552 & 0.3798 & 0.5651 \end{pmatrix}.$$

Based on P' after the normalization step, also the third assignment is identified correctly. The estimated profiles obtained by an adversary in a realistic scenario contain noise. The normalization step partially eliminates this noise yielding more reliable data.

The combination of these two effects allow the NSDA to de-anonymize messages with a higher success rate than the original SDA. As we show in Table 3.3, this attack runs faster than the PMDA for $t = 500$ and $t = 1000$, yet it achieves a lower success rate. It is a decision of the adversary which method suits her purposes best.

3.8 Conclusions

The main drawback of previously published practical Disclosure Attacks is their susceptibility to changes in the user behavioral model. Each of them seems to be optimized for a specific and restricted scenario. Our first contribution is a more general user behavior model, where the number of users' friends and the distribution of sending probabilities toward them is not restricted. Although this model is more flexible than previous proposals, it is not as versatile as one would desire and most probably far from real user behavior. More research needs to be performed on the influence of parameters like the users' sending rate or its variance over time on the effectiveness and efficiency of attacks in order to evaluate their impact on real anonymous communication networks.

In this chapter we have introduced the Perfect Matching Disclosure Attack (PMDA), that operates successfully without making assumptions on the users' preferences. Contrary to previous Disclosure Attacks it considers information about all senders participating in a round simultaneously, rather than focusing on individual users iteratively. This is bound to yield better results because it takes into account that all messages sent in a round are received only once (that is every input is liked to one, and only one, output). On the other hand focusing individually on users misses this information and as a result previous attacks can assign the same received message to several input messages. We have empirically compared it with previous work and have showed the advantage of our method when de-anonymizing messages.

A second advantage of the PMDA over previous work is its enhanced ability to estimate user profiles. Concerning a very restrictive user behavior model we empirically confirm that the PMDA yields a better separation of friends and non-friends than previous work. With respect to a general scenario we show that the PMDA reliably identifies users' friends when previously proposed methods fail. In our original paper [270], we also show how the enhanced profiles can be used as input for a new instance of de-anonymization. In fact, the PMDA can be chained as many times as desired, each time yielding a (slight) improvement over the outputs of the previous iteration.

Although we have shown that the PMDA is practical, it is computationally more expensive than previously proposed methods. Besides the fact that our proposal can be parallelized to a high degree to solve this problem, we have proposed a significantly sped-up variant, the Normalized Statistical Disclosure Attack. The NSDA is slightly less successful than the PMDA but it is almost as fast as the original SDA.

In this chapter we have focused on a basic anonymous channel as the threshold mix in order to better illustrate our attacks and their properties. However, realistic anonymous communication systems are expected to use more complex mixing algorithms. In the original paper [270] we show how the PMDA could be adapted to the scenario where a pool mix [192] is used. Further research is needed to validate our proposal.

Chapter 4

Bayesian inference to de-anonymize persistent communications

4.1 Introduction

We have seen in the previous chapter that mix-based systems do not protect the anonymity of users that persistently communicate with their contacts. After a number of messages are exchanged, the set of friends of each sender can be disclosed by a global passive adversary. The Perfect Matching Disclosure Attack, presented in previous chapter, allows to guess communication partners in a round of mixing with higher accuracy than its predecessors. Further, we have shown how this information can be in turn used to improve the estimation of users' sending profiles. However, we have seen that reusing information in a naïve manner to improve de-anonymization results, or to enhance the quality of the extracted profiles, is not optimal.

In this chapter we re-examine the problem of extracting profiles from traffic traces of anonymous communications and, in parallel, uncover who is talking with whom. We re-define the disclosure attack in anonymity systems [5,158,161], and analyze it using advanced Bayesian statistics. We note that at the heart of long-term traffic analysis lies an inference problem: from a set of public observations the adversary tries to infer a "hidden state" relating to who is talking to whom, as well as their long-term contacts. Applying Bayesian techniques provides a sound framework on which to build attacks using standard, well-studied algorithms to co-estimate multiple probabilities.

Contrary to previous analysis techniques, the samples output by Bayesian algorithms do not correspond to a specific inference, like "who is the most likely receiver of Alice's message?." In fact, they can be combined in many ways to infer arbitrary statements. For instance, the adversary may be interested in knowing whether Alice ever speaks to Bob, or if two target messages have the same originator. Besides their flexibility, Bayesian techniques output reliable error estimates, allowing the adversary to evaluate the confidence she can put on the results obtained. For instance, the algorithm can point at Charlie as Alice's most likely receiver. Nevertheless, it is not the same when the adversary is 100% certain of this assignment, than when her certainty is only fifty percent. The ability to obtain accurate error estimations allows the attacker to judge the quality of her inferences when operating in the wild.

Our key contributions are first a very general model to represent long-term attacks [5, 158, 161] against any anonymity system. This model generalizes the one presented in Chapter 3 by abstracting the concrete user behavior, and conceptually separating the application layer (i.e., the users' profiles) from the communication layer (i.e., the anonymous routing scheme). Second, we introduce the application of Bayesian inference techniques to the traffic analysis of anonymous communications. Throughout this chapter we show that our models and techniques lead to effective de-anonymization algorithms, while producing accurate error estimates. Furthermore they are far more flexible and reliable than previous ad hoc techniques.

The results presented in this chapter have been extracted from our original work *Vida: How to use Bayesian inference to de-anonymize persistent communications.* published at the *9th Privacy Enhancing Technologies Symposium* [79]. The techniques presented here are complemented by the ones introduced in the next chapter (originally published in [266]). The content of both chapters is extended in [78].

Chapter outline

The rest of this chapter is organized as follows: Sect. 4.2 offers an overview of Bayesian inference techniques, their relevance to traffic analysis, as well as an overview of the Gibbs sampling algorithm; Sect. 4.3 presents the Vida general model for anonymous communications, that can be used to model any system. In Sect. 4.4 we present a simplification of the model, the Vida Red-Blue model, that allows an adversary to perform inference on selected targets, as it would be operationally the case, along with an evaluation of the effectiveness of the inference technique. Finally we conclude in Sect. 4.5.

4.2 Bayesian inference and Monte Carlo methods

Bayesian inference is a branch of statistics with applications to machine learning and estimation [180]. Its key methodology consists in constructing a full probabilistic model of all variables in a system under study. Given observations of some of the variables, the model can be used to extract the probability distributions over the remaining, hidden, variables.

To be more formal let us assume that an abstract system consists in a set of hidden state variables $\mathcal{HS}$ and observations $\mathcal{O}$. We assign to each possible set of these variables a joint probability $\Pr[\mathcal{HS}, \mathcal{O}|\mathcal{C}]$ given a particular model $\mathcal{C}$. By applying Bayes rule we can find the distribution of the hidden state given the observations as:

$$\Pr[\mathcal{HS}, \mathcal{O}|\mathcal{C}] = \Pr[\mathcal{HS}|\mathcal{O}, \mathcal{C}] \cdot \Pr[\mathcal{O}|\mathcal{C}] \Rightarrow \Pr[\mathcal{HS}|\mathcal{O}, \mathcal{C}] = \frac{\Pr[\mathcal{HS}, \mathcal{O}|\mathcal{C}]}{\Pr[\mathcal{O}|\mathcal{C}]} \Rightarrow \tag{4.1}$$

$$\Pr[\mathcal{HS}|\mathcal{O}, \mathcal{C}] = \frac{\Pr[\mathcal{HS}, \mathcal{O}|\mathcal{C}]}{\sum_{\forall \mathcal{HS}} \Pr[\mathcal{HS}, \mathcal{O}|\mathcal{C}] \equiv \mathcal{Z}} = \frac{\Pr[\mathcal{O}|\mathcal{HS}, \mathcal{C}] \cdot \Pr[\mathcal{HS}|\mathcal{C}]}{\mathcal{Z}} . \tag{4.2}$$

The joint probability $\Pr[\mathcal{HS}, \mathcal{O}|\mathcal{C}]$ is decomposed into the equivalent $\Pr[\mathcal{O}|\mathcal{HS}, \mathcal{C}] \cdot \Pr[\mathcal{HS}|\mathcal{C}]$, describing the model and the a priori distribution over the hidden state. The quantity $\mathcal{Z}$ is simply a normalizing factor.

There are key advantages in using a Bayesian approach to inference that make it very suitable for traffic analysis applications:

- It provides a systematic approach to integrating all information available to an attacker, simply by incorporating all aspects of a system within the probability models [153].
- The problem of traffic analysis is reduced to building a generative model of the system under analysis. Knowing how the system functions is sufficient to encode and perform the attacks, and the inference details are, in theory, easily derived. In practice, computational limitations require carefully crafted models to be able to handle large systems.
- The output of the inference engine allows to compute probability distributions over all possible hidden states, not only the most probable solution as many other traffic analysis methods do (e.g., the attacks presented in the previous chapter).

The last point is the most important one: the probability distribution over hidden states given an observation, $\Pr[\mathcal{HS}|\mathcal{O}, \mathcal{C}]$, contains a lot of information about all possible states. The probability of error of particular aspects of the hidden state can be calculated to inform decision making. It is very different to assert that the most likely correspondent of Alice is Bob with certainty 99% than with certainty 5%. Extracting probability distributions over the hidden state allows us to compute such error estimates directly, without the need for an ad hoc analysis of false positives and false negatives. Furthermore, the analyst can use the inferred probability distribution to calculate directly anonymity metrics [86,238] as we will see in the next chapter.

Despite their power, Bayesian techniques come at a considerable computational cost. It is often not possible to compute or characterize directly the distribution $\Pr[\mathcal{HS}|\mathcal{O},\mathcal{C}]$ due to its complexity. In those cases, sampling-based methods are available to extract some of its characteristics. The key idea is that a set of samples $\mathcal{HS}_0,\ldots,\mathcal{HS}_\iota \sim \Pr[\mathcal{HS}|\mathcal{O},\mathcal{C}]$ are drawn from the a posteriori distribution, and used to estimate the marginal probability distributions of interest. For this purpose, Markov chain Monte Carlo methods have been proposed. These are stochastic techniques that perform a long random walk on a state space representing the hidden information, using specially crafted transition probabilities that make the walk converge to the target stationary distribution, namely $\Pr[\mathcal{HS}|\mathcal{O},\mathcal{C}]$. Once the Markov Chain has been built, samples of the hidden states of the system can be obtained by taking the current state of the simulation after a certain number of iterations.

4.2.1 Gibbs sampler

The Gibbs sampler [119] is a Markov chain Monte Carlo method to sample from joint distributions that have easy-to-sample marginal distributions. These joint distributions are often the a posteriori distribution resulting from the application of Bayes theorem, and thus Gibbs sampling has been extensively used to solve Bayesian inference problems. The operation of the Gibbs sampler is often referred to as *simulation*, but we must stress that it is unrelated to simulating the operation of the system under attack.

For illustration purposes we assume that an a posteriori distribution $\Pr[\mathcal{HS}|\mathcal{O},\mathcal{C}]$ can be written as a joint probability distribution $\Pr[X,Y|\mathcal{O},\mathcal{C}]$ that is difficult to sample directly. If, on the other hand, there is an efficient way of sampling from the marginal distributions $\Pr[X|Y,\mathcal{O},\mathcal{C}]$ and $\Pr[Y|X,\mathcal{O},\mathcal{C}]$, then Gibbs sampling is an iterative technique to draw samples from the joint distribution $\Pr[X,Y|\mathcal{O},\mathcal{C}]$. The algorithm starts at an arbitrary state (x_0,y_0). Then it iteratively updates each of the components through sampling from their respective distributions, i.e. $x_i \sim \Pr[X|Y=y_{i-1},\mathcal{O},\mathcal{C}]$, and $y_i \sim \Pr[Y|X=x_i,\mathcal{O},\mathcal{C}]$. After a sufficient number of iterations, the sample (x_i,y_i) is distributed according to the target distribution, and the procedure can be repeated to draw more samples. We stress that in this process the computation of the normalizing factor $\mathcal{Z}$ is not needed.

The other parameters of the Gibbs algorithm, namely the number of iterations necessary per sample, as well as the number of samples are also of some importance. The number of iterations has to be high enough to ensure the output samples are statistically independent. Calculating this number exactly is difficult so we use conservative estimates to ensure that we get good samples. The number of samples to be extracted, on the other hand, has an effect on the accuracy needed when estimating the marginal distributions, which can be achieved by running the sampler longer.

4.3 The Vida general Black-box model for anonymity systems

Long-term attacks traditionally abstract the internal functioning of any anonymity system and represent it as a black box, effectively operating as a very large threshold mix. This model has its limitations, and some studies have attempted to extend it. In this section we first propose the Vida Black-box model, the most flexible abstraction of an anonymity system so far, and base our Bayesian analysis on this model.

We start by proposing a ‘forward’ generative model describing how messages are generated and sent through the anonymity system. We then use Bayes rule to ‘invert’ the problem and perform inference on the unknown quantities. The broad outline of the generative model is depicted in Figure 4.1.

An anonymity system is abstracted as comprising N_{user} users that send a total of N_{msg}. Each user is associated with a sending profile Ψ_x describing how they select their correspondents when sending a message. We assume in this work that those profiles are simple multinomial distributions. We sample each user’s distribution independently to determine the receiver of each sent message. We denote the collection of all sending profiles by $\Psi = \{\Psi_x | x = 1 \ldots N_{\text{user}}\}$.

We consider that a sequence of senders $\text{Sen}_1, \ldots, \text{Sen}_{N_{\text{msg}}}$, out of the N_{user} users of the system, send a message while we observe the system. Using their sending profiles a corresponding sequence of receivers $\text{Rec}_1, \ldots, \text{Rec}_{N_{\text{msg}}}$ is selected to receive their messages. The probability of any receiver sequence is easy to compute. We denote this matching between senders and receivers as M:

$$\Pr[M|\Psi] = \prod_{x \in [1, N_{\text{msg}}]} \Pr[\text{Sen}_x \to \text{Rec}_x | \Psi_x] \,.$$

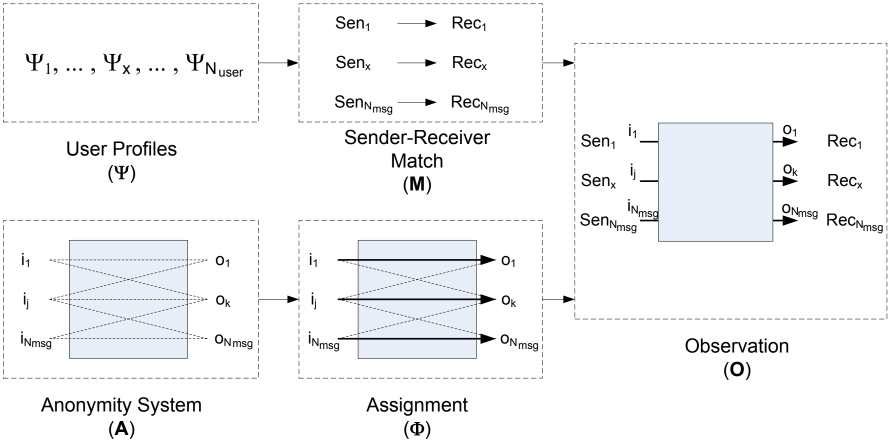

$\Psi_1, \ldots, \Psi_x, \ldots, \Psi_{N_{user}}$
User Profiles
(Ψ)
Sen_1 → Rec_1
Sen_x → Rec_x
$Sen_{N_{msg}}$ → $Rec_{N_{msg}}$
Sender-Receiver Match
(**M**)
Sen_1 i_1 o_1 Rec_1
Sen_x i_j o_k Rec_x
$Sen_{N_{msg}}$ $i_{N_{msg}}$ $o_{N_{msg}}$ $Rec_{N_{msg}}$
Observation
(**O**)
i_1 o_1
i_j o_k
$i_{N_{msg}}$ $o_{N_{msg}}$
Anonymity System
(**A**)
i_1 o_1
i_j o_k
$i_{N_{msg}}$ $o_{N_{msg}}$
Assignment
(Φ)

Figure 4.1: The generative model used for Bayesian inference in anonymous communications.

Besides the matching process where users choose their communication partners, an anonymity system $\mathcal{A}$ is used to route the messages. This anonymity system is abstracted as a bipartite graph linking input messages i_j with output messages o_k, regardless of the identity of their senders and receivers (see Sect. 3.4.2). We note that completeness of the bipartite graph is not required by the model. The edges of the bipartite graph are weighted with the probability of the input message i_j being output as o_k: $\Pr[i_j \to o_k|\mathcal{A}]$. In the previous chapter, the considered anonymity system $\mathcal{A}$ was a single threshold mix. When this mix is used, an input message i_j in a given round r is equally likely to correspond to any of the output messages o_k of that round, and has zero probability to be a message received in any other round. More formally, when a threshold mix handles t messages per round,

$$\Pr[i_j \to o_k|\mathcal{A}] = \begin{cases} 1/t & \text{if } i_j \in I_r,\, o_k \in O_r \\ 0 & \text{otherwise.} \end{cases}$$

This anonymity system $\mathcal{A}$ is used to determine a particular assignment of messages according to the weights $\Pr[i_j \to o_k|\mathcal{A}]$. A single perfect matching on the bipartite graph described by $\mathcal{A}$ is selected to be the correspondence between inputs and outputs of the anonymity system for a particular run of the anonymity protocol. We call this matching the assignment of inputs to outputs and denote it by Φ. Contrary to previous work [246] on probabilistic modeling, and following the strategy introduced in the previous chapter, we consider all inputs simultaneously. In this case, the probability of the assignment Φ is easy to calculate, given the set of all individual assignments $(i_j \to o_k)$:

$$\Pr[\Phi|\mathcal{A}] = \prod_j \frac{\Pr[i_j \to o_k|\mathcal{A}]}{\sum_{\text{free } i_l} \Pr[i_l \to o_k|\mathcal{A}]} \,.$$

This is simply the probability of the matching given the anonymity system weights. By free i_l we denote the set of sent messages i that has not yet been assigned an output message o as part of the match.

The assignment Φ of the anonymity system and the matching M of senders and receivers are combined to make up the observation of the adversary, denoted as $\mathcal{O}$. An adversary observes messages from particular senders Sen_x entering the anonymity system as messages i_j, and on the other side messages o_k exiting the network on their way to receivers Rec_y. No stochastic process takes place in this combination and therefore $\Pr[\mathcal{O}|M, \Phi, \Psi, \mathcal{A}] = 1$, given the choices of the users and the links between input and output messages the adversary's observation is uniquely defined.

Now that we have defined a full generative model for all the quantities of interest in the system, we turn our attention to the inference problem: the adversary observes $\mathcal{O}$ and knows the properties of the anonymity system $\mathcal{A}$, but is ignorant about the profiles Ψ, the matching M and the assignment Φ. We use Bayes theorem to calculate the probability $\Pr[M, \Phi, \Psi|\mathcal{O}, \mathcal{A}]$. We start with the joint distribution and solve for it:

$$\begin{aligned}
\Pr[\mathcal{O}, M, \Phi, \Psi|\mathcal{A}] &= \Pr[M, \Phi, \Psi|\mathcal{O}, \mathcal{A}] \cdot \Pr[\mathcal{O}|\mathcal{A}] \\
\Pr[\mathcal{O}, M, \Phi, \Psi|\mathcal{A}] &= \Pr[\mathcal{O}|M, \Phi, \Psi, \mathcal{A}] && (\equiv 1) \\
&\quad \cdot \Pr[M|\Phi, \Psi, \mathcal{A}] && (\equiv \Pr[M|\Psi]) \\
&\quad \cdot \Pr[\Phi|\Psi, \mathcal{A}] && (\equiv \Pr[\Phi|\mathcal{A}]) \\
&\quad \cdot \Pr[\Psi|\mathcal{A}]
\end{aligned}$$

$$\Rightarrow \Pr[M, \Phi, \Psi|\mathcal{O}, \mathcal{A}] = \frac{\Pr[M|\Psi]\Pr[\Phi|\mathcal{A}]}{\Pr[\mathcal{O}|\mathcal{A}] \equiv \mathcal{Z}} \Pr[\Psi|\mathcal{A}] \,.$$

We have discussed how to calculate the probabilities $\Pr[M|\Psi]$ and $\Pr[\Phi|\mathcal{A}]$. The quantity $\Pr[\Psi|\mathcal{A}] \equiv \Pr[\Psi]$ is the a priori belief the attacker has about user profiles and it is independent from the chosen anonymity system $\mathcal{A}$. We consider throughout our analysis that all profiles are a priori equally probable and reduce it to a constant $\Pr[\Psi] = c$. Taking into account those observations we conclude that the posterior probability sought is,

$$\Pr[M, \Phi, \Psi|\mathcal{O}, \mathcal{A}] \sim \prod_{x \in [1, N_{\text{msg}}]} \Pr[\text{Sen}_x \to \text{Rec}_x|\Psi_x] \cdot \prod_j \frac{\Pr[i_j \to o_k|\mathcal{A}]}{\sum_{\text{free } i_l} \Pr[i_l \to o_k|\mathcal{A}]} \,,$$

where we omit the constant normalizing factor $\Pr[\mathcal{O}|\mathcal{A}]$ as it is very hard to calculate. This restricts the methods we can use to compute the sought a posteriori distribution.

It is computationally infeasible to exhaustively enumerate the states of this distribution. Hence, to calculate the marginals of interest such as users' profiles, or likely recipients of specific messages, we have to resort to sampling states from that a posteriori distribution. Sampling directly is very hard (due to the interrelation between the profiles, the matches, and the assignments) hence Markov chain Monte Carlo methods are used.

4.3.1 A Gibbs sampler for the Vida Black-box model

Sampling states $(M_\iota, \Phi_\iota, \Psi_\iota) \sim \Pr[M, \Phi, \Psi|\mathcal{O}, \mathcal{A}]$ directly is hard, due to the complex interactions between the random variables. A Gibbs sampler significantly simplifies this process by only requiring us to sample from the marginal distributions of the random variables sought. Given an arbitrary initial state (Φ_0, Ψ_0), we can perform $\iota_{\max}$ iterations of the Gibbs algorithm as follows:

$$\begin{aligned} &\text{for } \iota := 1 \ldots \iota_{\max} : \\ \Phi_\iota, M_\iota &\sim \Pr[\Phi, M|\Psi_{\iota-1}, \mathcal{O}, \mathcal{A}] \\ \Psi_\iota &\sim \Pr[\Psi|\Phi_\iota, M_\iota, \mathcal{O}, \mathcal{A}] \,. \end{aligned}$$

Each of these marginal probabilities distributions is easy to sample:

- The distribution of assignments $\Pr[\Phi, M|\Psi_{\iota-1}, \mathcal{O}, \mathcal{A}]$ is subtle to sample directly. Each message assignment $i_j \to ok$ has to be sampled, taking into account that some message assignments are already taken by the time input message i_j is considered. For each input message i_j we sample an assignment o_k according to the distribution:

  $$i_j \to o_k \sim \Pr[i_j \to o_k|\text{free } o_k, \forall_{\text{assigned } o_v} i_v \to o_v, \mathcal{A}, \Psi] =$$
  $$= \frac{\Pr[i_j \to o_k|\mathcal{A}] \cdot \Pr[\text{Sen}_x \to \text{Rec}_y|\Psi_x]}{\sum_{\text{free } o_k} \Pr[i_j \to o_k|\mathcal{A}] \cdot \Pr[\text{Sen}_x \to \text{Rec}_y|\Psi_x]} \,.$$

  For complex anonymity systems $\mathcal{A}$, this algorithm might return only partial matches, when at some point an input message i_j has no unassigned candidate output message o_k left. Since we are only interested in perfect matchings, where all input messages are matched with different output messages, we reject such partial states and re-start the sampling of the assignment until a valid perfect matching is returned. This is effectively a variant of rejection sampling, to sample valid assignments.

The matchings between senders and receivers are uniquely determined by the assignments and the observations, so we can update them directly without any need for sampling, and regardless of the profiles (i.e. $M_\iota = f(\Psi_\iota, \mathcal{O})$).

- The distribution of profiles $\Pr[\Psi|\Phi_\iota, M_\iota, \mathcal{O}, \mathcal{A}]$ is straightforward to sample given the matching M_ι and assuming that individual profiles Ψ_x are multinomial distributions.

  We note that the Dirichlet distribution is a conjugate prior of the multinomial distribution, and we use it to sample profiles for each user. We denote as $\Psi_x = (\Pr[\mathrm{Sen}_x \to \mathrm{Rec}_1], \ldots, \Pr[\mathrm{Sen}_x \to \mathrm{Rec}_{N_{\mathrm{user}}}])$ the multinomial profile of user Sen_x. We also define a function that counts the number of times a user Sen_x is observed sending a message to user Rec_y in the match M, and denote it as $\mathrm{Ct}_M(\mathrm{Sen}_x \to \mathrm{Rec}_y)$. Sampling profiles $(\Psi_1, \ldots, \Psi_{N_{\mathrm{user}}}) \sim \Pr[\Psi|M]$ involves sampling independently each sender's profile Ψ_x separately from a Dirichlet distribution with the following parameters:

  $$\Psi_x \sim \mathrm{Dirichlet}(\mathrm{Ct}_M(\mathrm{Sen}_x \to \mathrm{Rec}_1) + 1, \ldots, \mathrm{Ct}_M(\mathrm{Sen}_x \to \mathrm{Rec}_{N_{\mathrm{user}}}) + 1)\,.$$

If the anonymity system $\mathcal{A}$ describes a simple bipartite graph, the rejection sampling algorithm described can be applied to sample assignments $i_j \to o_k$ for all messages. When this variant of rejection sampling becomes expensive, due to a large number of rejections, a Metropolis-Hastings [48] (see Sect. 5.2) based algorithm can be used to sample perfect matchings on the bipartite graph according to the distribution $\Pr[\Phi, M|\Psi_{\iota-1}, \mathcal{O}, \mathcal{A}]$. Our implementation was tested against mix-based anonymity systems, with bipartite graphs representing the anonymity system that do not lead to any rejections.

The Gibbs sampler can be run multiple times to extract multiple samples from the a posteriori distribution $\Pr[M, \Phi, \Psi|\mathcal{O}, \mathcal{A}]$. Instead of restarting the algorithm at an arbitrary state (M_0, Φ_0, Ψ_0), it is best to set the starting state to the last extracted sample, that is likely to be within the typical set of the distribution. This speeds up convergence to the target distribution.

4.4 A computationally simple Vida Red-Blue model

As showed in the previous chapter, co-estimating sender profiles with the assignments has some advantages, and our Bayesian analysis so far reflects this approach. Senders are associated with multinomial profiles with which they choose specific correspondents. We sample these profiles using the Dirichlet distribution, and use them to directly sample weighted perfect assignments in the anonymity system. The output of the algorithm is a set of samples of the hidden state, that allows the adversary to estimate the marginal distributions of specific senders sending to specific receivers.

We note that this approach is very general, and might go beyond the day-to-day needs of a real-world adversary. An adversary is likely to be interested in particular target senders or receivers, and might want to answer the question: "who has sent this message to Bob?" or "who is friends with Bob?." We present the Vida Red-Blue model to answer such questions, which is much simpler, both mathematically and computationally, than the general Vida model presented in the previous sections.

Consider that the adversary chooses a target receiver Bob (that we call "Red"), while ignoring the exact identity of all other receivers and simply tagging them as "Blue." The profiles Ψ_x of each sender can be collapsed into a simple binomial distribution describing the probability sender x sends to Red or to Blue. It holds that:

$$\Pr[\mathrm{Sen}_x \rightarrow \mathrm{Red}|\Psi_x] + \Pr[\mathrm{Sen}_x \rightarrow \mathrm{Blue}|\Psi_x] = 1\,. \tag{4.3}$$

Matchings M map each observed sender of a message to a receiver class, either Red or Blue. Given the profiles Ψ the probability of a particular match M is:

$$\Pr[M|\Psi] = \prod \Pr[\mathrm{Sen}_x \rightarrow \mathrm{Red}\ /\ \mathrm{Blue}|\Psi_x]\,.$$

The real advantage of the Vida Red-Blue model is that different assignments Φ now belong to equivalence classes, since all Red or Blue receivers are considered indistinguishable from each other [121]. In this model the assignment bipartite graph can be divided into two sub-graphs: the sub-graph Φ_R contains all edges ending on the Red receiver (as she can receive more than one message in a mixing round), while the sub-graph Φ_B contains all edges ending on a Blue receiver. We note that these sub-graphs are complementary and any of them uniquely defines the other. The probability of each Φ can then be calculated as:

$$\Pr[\Phi|\mathcal{A}] = \sum_{\forall \Phi_B} \Pr[\Phi_B, \Phi_R|\mathcal{A}] = \sum_{\forall \Phi_B} \Pr[\Phi_B|\Phi_R, \mathcal{A}] \cdot \Pr[\Phi_R|\mathcal{A}] =$$

$$= \Pr[\Phi_R|\mathcal{A}] \cdot \sum_{\forall \Phi_B} \Pr[\Phi_B|\Phi_R, \mathcal{A}] = \Pr[\Phi_R|\mathcal{A}]\,.$$

The probability of an assignment in an equivalence class defined by the assignment to Red receivers, only depends on Φ_R describing this assignment. The probability of assignment Φ_R can be calculated analytically as:

$$\Pr[\Phi_R|\mathcal{A}] = \prod_{j \in \Phi_R} \frac{\Pr[i_j \to o_k]}{\sum_{\text{free } i_l} \Pr[i_l \to o_k]} \,.$$

The assignment Φ_R must be a sub-graph of at least one perfect matching on the anonymity system $\mathcal{A}$, otherwise the probability becomes $\Pr[\Phi|\mathcal{A}] = 0$. As for the full model the probability of all the hidden quantities given the observation is:

$$\Pr[M, \Phi, \Psi|\mathcal{O}, \mathcal{A}] = \frac{\Pr[M|\Psi] \Pr[\Phi_R|\mathcal{A}]}{\Pr[\mathcal{O}|\mathcal{A}] \equiv \mathcal{Z}} \Pr[\Psi|\mathcal{A}] \,. \tag{4.4}$$

The a priori probability over profiles $\Pr[\Psi|\mathcal{A}]$ is simply a prior probability over the parameters of a binomial distribution. Each profile can be distributed as $\Pr[\Psi_x|\mathcal{A}] = \text{Beta}(1, 1)$, equivalent to a standard uniform distribution, if nothing is to be assumed about the sender's Sen_x relationship with the Red receiver.

In practice a prior distribution $\Pr[\Psi_x|\mathcal{A}] = \text{Beta}(1, 1)$ is too general, and best results are achieved by using a prior supporting skewed distributions, such as $\text{Beta}(1/100, 1/100)$. This reflects the fact that social ties are a priori either strong or non existent. Given enough evidence the impact of this choice of prior fades quickly away.

4.4.1 A Gibbs sampler for the Vida Red-Blue model

Implementing a Gibbs sampler for the Vida Red-Blue model is very simple. The objective of the algorithms is, as for the general model, to produce samples of profiles (Ψ_ι), assignments and matches (Φ_ι, M_ι) distributed according to the Bayesian a posteriori distribution $\Pr[M, \Phi, \Psi|\mathcal{O}, \mathcal{A}]$ described by Eq. 4.4.

The Gibbs algorithm starts from an arbitrary state (Ψ_0, Φ_0) and iteratively samples new marginal values for the profiles $(\Phi_\iota, M_\iota \sim \Pr[\Phi, M|\Psi_{\iota-1}, \mathcal{O}, \mathcal{A}])$ and the valid assignments $(\Psi_\iota \sim \Pr[\Psi|M_\iota, \Phi_\iota, \mathcal{O}, \mathcal{A}])$. The full matchings are a deterministic function of the assignments and the observations, so we can update them directly without any need for sampling (i.e. $M_\iota = f(\Psi_\iota, \mathcal{O})$).

As for the general Gibbs sampler, sampling from the desired marginal distributions can be done directly. Furthermore the Vida Red-Blue model introduces some simplifications that speed up inference:

- **Sampling assignments.** Sampling assignments of senders to Red nodes (i.e. $\Phi_{R\iota}, M_\iota \sim \Pr[\Phi, M|\Psi_{\iota-1}, \mathcal{O}, \mathcal{A}]$) can be performed by adapting the rejection sampling algorithm presented for the general model. The key modification is that only assignments to Red receivers are of interest, and only an arbitrary assignment to blue receivers is required (to ensure such an assignment exists). This time for each Red output messages ok we sample an input message i_j according to the distribution:

$$i_j \to o_k \sim \Pr[i_j \to o_k | \text{free } i_j, \forall_{\text{assigned } i_v} i_v \to o_v, \mathcal{A}, \Psi] =$$

$$= \frac{\Pr[i_j \to o_k|\mathcal{A}] \cdot \Pr[\text{Sen}_x \to \text{Red}|\Psi_x]}{\sum_{\text{free } i_l} \Pr[i_l \to o_k|\mathcal{A}] \cdot \Pr[\text{Sen}_l \to \text{Red}|\Psi_x]} .$$

- **Sampling profiles.** Sampling a profile $\Psi_\iota \sim \Pr[\Psi|M_\iota, \Phi_\iota, \mathcal{O}, \mathcal{A}]$ for every user x simply involves drawing a sample from a Beta distribution with parameters related to the number of links to Blue and Red receivers. To be formal we define a function $\text{Ct}_M(\text{Sen}_x \to \text{Red, Blue})$ that counts the number of messages in a match that a user x sends to a Red or Blue receiver. The profile of user x is then sampled as:

  $$\Psi_x \sim \text{Beta}(\text{Ct}_M(\text{Sen}_x \to \text{Blue}) + 1, \text{Ct}_M(\text{Sen}_x \to \text{Red}) + 1) .$$

  This yields a binomial parameter that is the profile of user x, describing the probability they send a message to a Red target user.

The cost of each iteration is proportional to sampling N_{user} Beta distributions, and sample from the distribution of senders of each of the Red messages. Both the sampling of profiles, and the sampling of assignments can be performed in parallel, depending on the topology. In case a large number of samples are needed multiple Gibbs samplers can be run on different cores or different computers to produce them.

4.4.2 Evaluation

The Vida Red-Blue model for inferring user profiles and assignments was evaluated against synthetic anonymized communication traces, to test its effectiveness. The communication traces include messages sent by up to 1000 senders to up to 1000 receivers. Each sender is assigned 5 contacts at random, to whom they send messages with equal probability. Messages are anonymized in discrete rounds using a threshold mix that gathers 100 messages before sending them to their receivers as a batch.

The generation of communication patterns was peculiar to ensure a balance between inferring the communications of a target user (as in the traditional disclosure, hitting set and statistical disclosure attacks) to a designated Red receiver, as well as to gain enough information about other users to build helpful

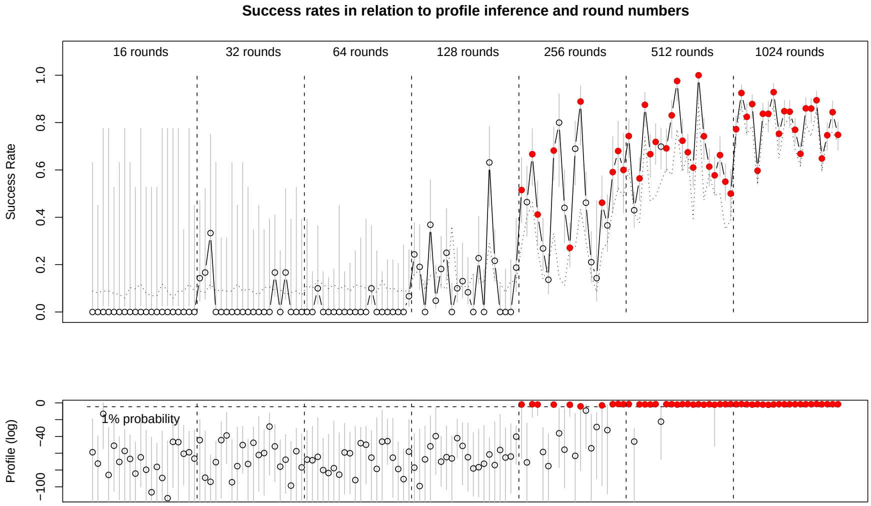


Figure 4.2: Performance of the Vida Red-Blue model in assigning senders to the target red receiver, as a function of the number of rounds observed. Twenty sample experiments are used per round number.

profiles for them. A target sender was included in 20% of the rounds, and the Red node was chosen to be one of their friends. A sequence of experiments were performed to assess the accuracy of the attack after observing an increasing number of rounds of communication.

The aim of each experiment is to use the samples returned by a Gibbs sampler implementing the Vida Red-Blue model to guess the sender of each message that arrives at a designated Red receiver. The optimal Bayes criterion [46] is used to select the candidate sender of each Red message: the sender with the highest a posteriori probability is chosen as the best candidate. This probability is estimated by counting the number of times each user were the sender of a target Red message in the samples returned by the Gibbs algorithm. The Bayesian probability of error, i.e. the probability another sender is responsible for the Red message, is also extracted, as a measure of the certainty of each of these “best guesses.” For each experiment the Gibbs sampler was used to extract 200 samples, each using 100 iterations of the Gibbs algorithm. The first 5 samples were discarded, to ensure stability is reached before drawing any inferences.

A summary of the results for each experiment is presented in Figure 4.2. The top graph illustrates the fraction of correct guesses per experiment (on the x axis –

we selected 20 random experiments to display per round number) grouped by the number of rounds of communication observed (16, 32, 64, 128, 256, 512 and 1024). For each experiment the fraction of correctly identified senders is marked by a circle, along with its 90% confidence interval. The dashed line of the same graph represents the prediction of success we get from the Bayesian inference engine. The bottom graph on Figure 4.2 illustrates, for each of the experiments, on a logarithmic scale the inferred probability with which the target sender chooses the Red node as recipient. We also plot the 50% confidence interval over the profile inference. The solid circle on both graphs represents that the inferred probability that the target sends to the red receiver is high (median greater than 1%).

Some key conclusions emerge from the experiments illustrated on Figure 4.2:

- The key trend we observe is, as expected, that the longer the observation in terms of rounds, the better the attack. Within 1024 rounds we expect the target sender to have sent only about 40 messages to the designated Red target. Yet, the communication is traced to them on average 80% of the cases with high certainty. Even when only 256 rounds are observed the correct assignment is guessed in about 50% of the time.
- The quality of the inference when it comes to the correspondence between messages, senders and receivers, is intimately linked to the quality of the profile inference. The solid circles mark experiments in which the inference process indicates that the median value for the probability the target sender is friends with the target Red receiver is high (greater than 1%). We observe that these experiments are linked to high success rates when it comes to linking individual messages to the target sender. We also observe that the converse is true: insufficient data leads to poor profiles, that in turn lead to poor predictions about communication relationships (e.g., 16 or 32 rounds observed).
- The probability of success estimates (represented on the top graph by a dotted line) predict well the success rate of the experiments. Our prediction systematically falls within the 90% confidence interval of the estimated error rate. This shows that the Vida Red-Blue model is a good representation of the process that generated the traces and thus the estimates coincide with the actual observed error rate, on average. This is due to the very general model for Vida Red-Blue profiles that represent reality accurately after a few rounds. Yet, when few rounds are observed the a priori distribution of profiles dominates the inference, and affects the error estimates.

A key question is how the results from the Vida Red-Blue model compare with traditional traffic analysis attacks, like the Statistical Disclosure Attack (SDA [60, 183], see Sect. 3.3), the Normalized Statistical Disclosure Attack (NSDA [270], see Sect. 3.7) or the Perfect Matching Disclosure Attack (PMDA [270], see Sect. 3.4).

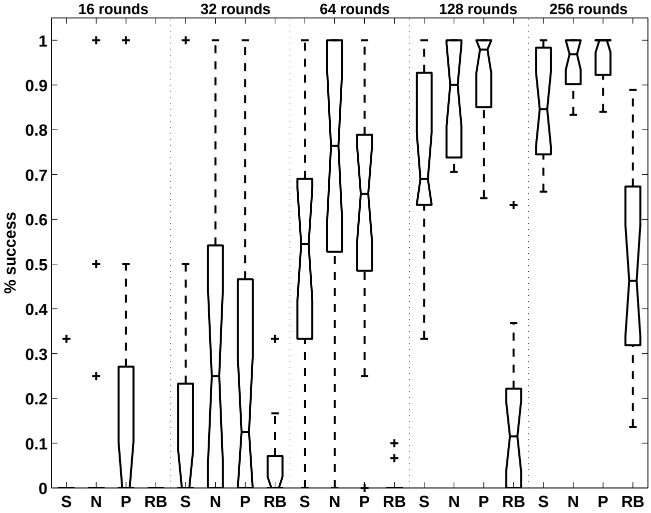


Figure 4.3: Performance of the Vida Red-Blue inference model (RB) compared to the SDA (S), NSDA (N) and PMDA (P).

The SDA attack simply uses first order frequencies to guess the profiles of senders. It is fast but inaccurate. The NSDA constructs a traffic matrix from senders to receivers, that is normalized to be doubly stochastic. The operation is as fast as matrix multiplication, and yields very good results. The PMDA finds perfect matchings between senders and receivers based on a rough profile extraction step – it is quite accurate but slow.

Figure 4.3 illustrates the relative performances of the different attacks compared with the Vida Red-Blue model proposed. We observe that the inference based technique is worse than the SDA, and performs much worse than the NSDA and PMDA in most settings. This is due to our strategy for extracting best estimates for the senders: we use the output samples to chose the sender with highest marginal probability instead of extracting a full match with the maximal marginal probability. In that sense applying an algorithm to find the maximal perfect matching based on the marginal probabilities output by the RB attacks should produce much better results.

Despite the lower success rate inference-based techniques can be advantageous. Their key strength is the certainty that no systematic bias has been introduced by using data twice, as reported in [89, 270], and the tangible and reliable error estimate they output. A traffic analyst is thus able to judge the quality of the inference.

A second important advantage is the ability to infer who is the “second most likely" receiver, compute anonymity metrics, or other arbitrary statements on the a posteriori probability distribution of profiles and assignments. This can be done efficiently simply using the samples output by the Gibbs algorithm. Furthermore the correct probabilities of error can be associated with those probabilistic statements.

4.5 Conclusions

The contribution of this chapter is two-fold: first it presents Vida, the first truly general model for abstracting any anonymity system, in the long term, to perform de-anonymization attacks. Users and their preferences are modeled in the most general way, using multinomial profiles, eliminating the need to know the number of contacts each sender has. Instead of abstracting an anonymity system as a single threshold mix, or even pool mix, an arbitrary weighted mapping of input to output messages can be used. We show that the model performs well when it comes to guessing who is talking to whom, as well as guessing the profiles of senders. Further, we presented the Vida Red-Blue model. It allows the traffic analyst to focus on specific targets instead of dealing with the full system. Additionally, the Vida Red-Blue model has the potential to be implemented efficiently and parallelized aggressively.

The second contribution is methodological. We have shown how Bayesian inference, and in particular Markov chain Monte Carlo sampling, is an appropriate framework to evaluate the resistance of an anonymity system to traffic analysis attacks. It allows the analyst to re-use information while ensuring that no systematic bias is introduced, as occurred in the enhanced profiling methodology we introduced in Chapter 3. Our method indicates a clear way to start the analysis with the definition of a probabilistic model that defines the likelihood of inputs corresponding to outputs (respectively senders communicating with receivers). This model is later inverted by applying the Bayes rule in order to find a probability distribution easier to sample from such that the analyst can infer quantities of interest. These quantities can answer arbitrary questions about the events in the system. As opposed to previous work in which just the question "who is the most likely receiver of Alice's message?" could be answered, other statements, for instance "have Alice and Bob communicated?" can be evaluated. Further, the method outputs reliable error estimates for these inferences that allow the analyst to evaluate the confidence she can have in the results obtained.

In this chapter we have performed a first step in the exploration of the applicability of inference techniques to problems in traffic analysis – in the hope that it eventually outperforms established techniques. Some future directions include the definition of better user models, the analysis of the internals of anonymity systems (started in the next chapter), as well as a better integration of prior information and learning. The inference approach leans itself well to be extended to encompass these problems, that have in the past been a thorn on the side of traffic analysis techniques.

Chapter 5

A Bayesian framework for the analysis of anonymous communication systems

5.1 Introduction

In the previous chapter we have shown how Bayesian inference can be used to extract communication profiles and uncover communication partners from traffic traces of anonymous communications systems. In order to illustrate our techniques, we abstracted the anonymity network $\mathcal{A}$ as an opaque threshold mix and considered that all inputs are equally likely to correspond to any outgoing message. However, we cannot expect that real anonymity systems can be abstracted in this way. Routing constraints or background knowledge of the adversary may bias these probabilities, changing the result of the analysis.

Extracting probability distributions over possible receivers of messages in an anonymity system, subject to constraints on its functioning and the observations of an adversary, is also the basis to compute anonymity metrics. Measures of anonymity based on information theory and decision theory were proposed [52,81, 86,238,263] to quantify the security of such systems. Although very popular, these metrics are difficult to apply in the presence of constraints that deployed systems impose, since the exact calculation of the required distributions is an intractable problem (as pointed out by Serjantov [237]).

Our key contribution is a framework to estimate, with arbitrarily high accuracy, the distributions necessary for computing a wide variety of anonymity metrics for relay-based mix networks. We achieve this by casting the problem of extracting these distributions as a probabilistic inference problem, and solve it using established Bayesian inference frameworks. In particular, in this chapter we consider the Metropolis-Hastings algorithm, another Markov chain Monte Carlo (MCMC) sampling technique similar to the Gibbs sampler introduced in the previous chapter.

Our analysis of mix networks incorporates most aspects and attacks previously presented in the literature: constraints on paths length, node selection [238], bridging and fingerprinting attacks [77], social relations of users [89], and erratic user behavior. For the first time, all these aspects are brought under a common framework allowing the adversary to combine them all when the analyzing a system. Further extensions to describe other aspects of mix networks can also be accommodated. This is the most comprehensive and flexible model of a mix-based anonymity network so far.

The Bayesian traffic analysis techniques presented have two key advantages. First, they allow optimal use of all information when drawing conclusions about who is talking to whom. Second, they provide the analyst with an a posteriori probability over all scenarios of interest, whereas previous attacks only provided the most likely candidate solution. The evaluation of our work focuses on establishing the correctness of those distributions.

The results presented in this chapter have been extracted from our original work *The Bayesian Analysis of Mix Networks.* published at the *16th ACM Conference on Computer and Communications Security (CCS 2009)* [266]. The techniques presented here complement the results introduced in the previous chapter (originally published in [79]). The content of both chapters is extended in [78].

Chapter outline

The chapter is organized as follows: we present a brief overview of the Metropolis-Hastings algorithm, a Markov chain Monte Carlo method in Sect. 5.2. Section 5.3 describes a generic probabilistic model of a mix network, and Sect. 5.4 shows how to build a Metropolis-Hastings-based engine to infer its hidden state. The correctness and accuracy of the inference engine is studied in Sect. 5.5, and Sect. 5.6 explains how to use the output of the sampler to compute anonymity. Finally, we discuss some future directions and conclusions in Sect. 5.7.

5.2 The Metropolis-Hastings algorithm

The Metropolis-Hastings (MH) algorithm [136] is a Markov chain Monte Carlo method that can be used to sample from arbitrary distributions (see Sect. 4.2 for further information on Bayesian inference and MCMC methods). It operates by performing a long random walk on a state space representing the hidden information, using specially crafted transition probabilities that make the walk converge to the target stationary distribution, namely $\Pr[\mathcal{HS}|\mathcal{O},\mathcal{C}]$. Its operation is often referred to as *simulation*, but we must stress that it is unrelated to simulating the operation of the system under attack.

The MH algorithm's key state is a single instance of the hidden state, called the *current state* and denoted $\mathcal{HS}_\iota$. Given the current state, a *candidate state* $\mathcal{HS}'$ is selected according to a probability distribution $Q(\mathcal{HS}'|\mathcal{HS}_\iota)$. A value α is defined as:

$$\alpha = \frac{\Pr[\mathcal{HS}'|\mathcal{O},\mathcal{C}] \cdot Q(\mathcal{HS}_\iota|\mathcal{HS}')}{\Pr[\mathcal{HS}_\iota|\mathcal{O},\mathcal{C}] \cdot Q(\mathcal{HS}'|\mathcal{HS}_\iota)}\,. \tag{5.1}$$

When $\alpha \geq 1$, the candidate state is accepted as the current state, otherwise it is only accepted with probability α. This process is repeated multiple times, and after a certain number of iterations δ the current state is output as a sample $\mathcal{HS}_{\iota+1}$. More samples can be extracted by repeating this process. It must be taken into account that before collecting the first sample we must wait a *burn-in* period until the sampler converges and visits states according to the probability distribution sought. The pseudocode representing the Metropolis-Hastings operation is shown in Algorithm 1.

The algorithm is very generic, and can be used to sample from any distribution on any state space, using custom transition probabilities Q. It is particularly interesting that the distribution Q used to propose new candidates can be arbitrary without affecting the correctness of the process, as long as both $Q(\mathcal{HS}'|\mathcal{HS}_\iota) > 0$ and $Q(\mathcal{HS}_\iota|\mathcal{HS}') > 0$, and the Markov Chain it forms fully connects all hidden states and it is ergodic.[1] Despite the apparent freedom in choosing the distribution Q, in practise it must be easy to compute and sample, and be fast mixing to reduce the number of iterations between independent samples. Since the probabilities $\Pr[\mathcal{HS}'|\mathcal{O},\mathcal{C}]$ and $\Pr[\mathcal{HS}_\iota|\mathcal{O},\mathcal{C}]$ need to only be known up to a multiplicative constant to calculate α, we do not need to know the normalizing factor $\mathcal{Z}$ (see Eq. 4.1).

As for the Gibbs algorithm presented in the previous chapter, the number of samples and the number of iterations necessary to get them are of some importance

[1] Connection and ergodicity are needed in order to ensure that all states can be visited, that they can be visited more than once, and that the samples output are independent of the initial state chosen to start the inference engine.

Algorithm 1 Metropolis-Hastings algorithm

```
$\mathcal{HS}_\iota$ // arbitrary initial state
$cnt = 0$ // sampler iterations counter
$s = 0$ // samples output counter
while $s \neq N_{\mathrm{MH}}$ do
    sample $\mathcal{HS}' \sim Q(\mathcal{HS}'|\mathcal{HS}_\iota)$ //propose candidate state $\mathcal{HS}'$
    compute $Q(\mathcal{HS}'|\mathcal{HS}_\iota)$ {probability of proposing $\mathcal{HS}'$ departing from $HS_\iota$}
    compute $Q(\mathcal{HS}_\iota|\mathcal{HS}')$ {probability of proposing $\mathcal{HS}_\iota$ departing from $HS'$}
    compute $\alpha$ {as in Eq. 5.1}
    if $\alpha \geq 1$ then
        $\mathcal{HS}_{\iota+1} = \mathcal{HS}'$ {the proposed state becomes the current state}
    else
        $\mathcal{HS}_{\iota+1} = \mathcal{HS}_\iota$ {the current state states unchanged}
    end if
    if $cnt \pmod{\delta}$==0 & $cnt$ >burn-in then
        store $\mathcal{HS}_{\iota+1}$ as a sample
        $s = s + 1$ {increase the counter of samples}
    end if
    $cnt = cnt + 1$ {increase the counter of iterations}
end while
```

for the correctness of the inferences. We choose the number of iterations of the MH algorithm experimentally such that the output samples are independent; and collect enough samples to demonstrate the utility of our techniques. We recall that the number of MH samples increases the accuracy of the marginal distributions that are estimated. Higher accuracy can be achieved by running the sampler longer than in our experiments.

The MH method can be run in parallel on multiple processors, cores, or a distributed cluster: all processes output samples that can be aggregated and analyzed centrally. Our experiments made use of this property on a multi-core two processor machine.

5.3 The mix network model

The first step to perform Bayesian inference is to define a probabilistic model that describes all observations and hidden states of a system. In this section, we present such a model for a set of users sending messages over a mix network to a set of receivers. The model includes traditional aspects of mix networks, e.g. path length constraints, and further incorporates incomplete observations, erratic clients, bridging attacks, and social network information (who is friends with whom, relationships' strength, etc.).

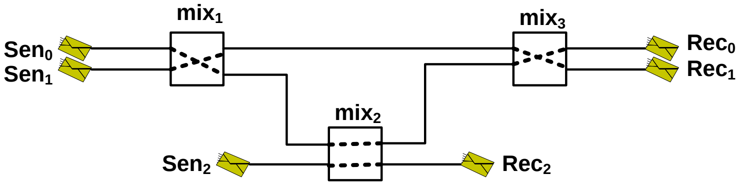


Figure 5.1: Observation of the network and Hidden State

We consider an anonymity system formed by N_{mix} threshold mixes [47] through which a population of N_{user} users sends messages. When sending a message, a user selects a receiver amongst her set of contacts and a path in the network to route the message. The path is determined by the preferences of the user and a set of constraints $\mathcal{C}$ imposed by the system (e.g., maximum path length, restrictions on the choice of mixes, etc). We denote the sender of an incoming message to the system i_j as Sen_j and the receiver of an outgoing message from the system o_k as Rec_k.

In order to carry out our analysis we observe the system over a period of time from T_0 to $T_{\max}$ (assuming that all mixes are empty at T_0). During this period, N_{msg} messages traveling through the system are monitored by a passive adversary, generating an *Observation* ($\mathcal{O}$). This observation is formed by records of communications between the entities (senders, mixes, and receivers).

Our goal is to determine the *probability that a message entering the network corresponds to each of the messages leaving it*, given an observation $\mathcal{O}$. This is equivalent to determining the correspondence between inputs and outputs in each of the mixes. We call the collection of the input-output relationships of all mixes the *Hidden State* of the system, and denote it as $\mathcal{HS}$.

Figure 5.1 depicts an instance of a system where 3 users send 3 messages through a network formed by 3 threshold mixes with threshold $t = 2$. In this setting a passive observer can monitor the following events ($x \rightarrow y$ denotes entity x sending a message to entity y) and construct an observation $\mathcal{O}$ with them:

$$\mathcal{O} = \{ \begin{array}{lll} \text{Sen}_0 \rightarrow \text{mix}_1 \ , & \text{mix}_1 \rightarrow \text{mix}_3 \ , & \text{mix}_2 \rightarrow \text{Rec}_2 \ , \\ \text{Sen}_1 \rightarrow \text{mix}_1 \ , & \text{mix}_1 \rightarrow \text{mix}_2 \ , & \text{mix}_3 \rightarrow \text{Rec}_0 \ , \\ \text{Sen}_2 \rightarrow \text{mix}_2 \ , & \text{mix}_2 \rightarrow \text{mix}_3 \ , & \text{mix}_3 \rightarrow \text{Rec}_1 \ \} \end{array}$$

These events are represented with solid lines in Fig. 5.1. A possible $\mathcal{HS}$ (correspondences between incoming and outgoing messages at all mixes) for this instance is represented with dashed lines.

Given an observation and a hidden state we define a path P_j for each of the messages i_j entering the network, representing its trajectory through the system. A path consists of a series of observed events that are linked by the relations stated in the hidden state. In the example, message i_1 follows the path $P_1 = \{\mathrm{Sen}_1 \to \mathrm{mix}_1, \mathrm{mix}_1 \to \mathrm{mix}_3, \mathrm{mix}_3 \to \mathrm{Rec}_1\}$. We note that a set of paths $\mathcal{P} = \{P_j,\, x = 1, \ldots, N_{\mathrm{msg}}\}$ *uniquely* determines an observation and a hidden state. Hence, given a set of constraints $\mathcal{C}$, their probability should be strictly equal, i.e. $\Pr[\mathcal{P}|\mathcal{C}] = \Pr[\mathcal{O}, \mathcal{HS}|\mathcal{C}]$. By applying Bayes theorem we can relate the probability of a hidden state (that we are trying to infer) to the observations and the paths that it forms as $\Pr[\mathcal{HS}|\mathcal{O}, \mathcal{C}] = \Pr[\mathcal{P}|\mathcal{C}]/\mathcal{Z}$ where $\mathcal{Z}$ is a normalizing constant.

Proof: Given that the probability of a path is restricted by the constraints $\mathcal{C}$ and that $\Pr[\mathcal{O}, \mathcal{C}]$ is a constant we obtain the following equation:

$$\Pr[\mathcal{P}|\mathcal{C}] = \Pr[\mathcal{O}, \mathcal{HS}|\mathcal{C}] = \Pr[\mathcal{HS}|\mathcal{O}, \mathcal{C}] \cdot \Pr[\mathcal{O}, \mathcal{C}]$$

$$\Rightarrow \Pr[\mathcal{HS}|\mathcal{O}, \mathcal{C}] = \frac{\Pr[\mathcal{O}, \mathcal{HS}|\mathcal{C}]}{\Pr[\mathcal{O}, \mathcal{C}]} = \frac{\Pr[\mathcal{O}, \mathcal{HS}|\mathcal{C}]}{\sum_{\mathcal{HS}} \Pr[\mathcal{HS}, \mathcal{O}|\mathcal{C}] \equiv \mathcal{Z}} = \frac{\Pr[\mathcal{P}|\mathcal{C}]}{\mathcal{Z}} \; \square \tag{5.2}$$

Hence, we can say that sampling hidden states $\Pr[\mathcal{HS}|\mathcal{O}, \mathcal{C}]$ is equivalent to sampling paths $\Pr[\mathcal{P}|\mathcal{C}]$ using Bayesian inference techniques. In the next sections we present a probability model of paths under different system-based and user-based constraints. Ultimately, we describe how to use the Metropolis-Hastings method to sample from that model.

We must stress that we have arbitrarily chosen the probability distributions that are used in this thesis to describe the mix network constraints. The model we present is, however, flexible enough to accommodate any other distribution instead. The analyst that wants to apply our analysis methods must make sure that the probability distributions he chooses actually represent the network under study. Note that routing constraints are in general easy to model as they are encoded in the software used by the clients, following public algorithms. Social characteristics, however, are more difficult to model as they are specific to users and change over time. This difficulty can be overcome by inferring them in parallel with correspondences amongst input and output messages as described in the previous chapter.

5.3.1 Basic constraints

First, we present our model for *basic* constraints concerning the user's choice of mixes to relay messages and the length of the path.

We assume that the system allows the user to choose paths of length $L_j, L_j = L_{\min}, \ldots, L_{\max}$. We consider that the user selects this length uniformly at random amongst the possible values. There is nothing special about the uniform distribution of path lengths, and an arbitrary distribution can be used instead. The probability of path P_j being of length l is:

$$\Pr[L_j = l|\mathcal{C}] = \frac{1}{L_{\mathrm{max}} - L_{\mathrm{min}} + 1} \,,$$

that is, the probability of choosing a length between L_{min} and L_{max} given that any length is equally likely.

Once the length is determined, the user has to choose the mixes on the path. We consider any sequence of mixes of the chosen length as equally likely, with the only condition that mixes have to be distinct. The possible ways in which the l mixes forming a path can be chosen is given by the permutations of length l out of the N_{mix} mixes forming the system. Thus, the probability of choosing a sequence Ω_j of l distinct mixes is:

$$\Pr[\Omega_j|L_j = l, \mathcal{C}] = \frac{1}{C(N_{\mathrm{mix}}, l)/P(l)} = \frac{P(l)}{C(N_{\mathrm{mix}}, l)} = \frac{(N_{\mathrm{mix}} - l)!}{N_{\mathrm{mix}}!} \,.$$

Assuming that the choice of the length of a path and the choice of mixes belonging to it are independent, the probability of selecting a path P_j formed by the l mixes in Ω_j is:

$$\Pr[P_j|\mathcal{C}] = \Pr[L_j = l|\mathcal{C}] \cdot \Pr[\Omega_j|L_j = l, \mathcal{C}] \cdot I_{\mathrm{set}}(P_j) \,, \tag{5.3}$$

where the last element represents an indicator of the choice of mixes being a set or a multiset. This indicator takes value 1 when all mixes in the path are different, 0 otherwise.

Since the observation is limited in time, it may be the case that some messages enter the network during the observation period but have not left it when the period ends. This happens when messages enter mixes that do not receive enough inputs to flush, and thus stay in those mixes at the end of the observation. For these messages, it is not possible to derive the choices of the user in terms of path length and mixes, as we only have a partial observation of the path. Such an example is shown in Fig. 5.2, representing an instance of a network formed by threshold mixes ($t = 2$) in which users can choose paths of length $L_j \in [2, 3]$. The message sent by Sen_2 arrives at mix_4, but it is never forwarded to any other mix or to its recipient since no more messages are received by this mix (this trajectory is shown in lighter gray in the Fig. 5.2). At this point, an adversary cannot assume Sen_2 chose $L_2 = 2$ and must consider the possibility that the choice could be $L_2 = 3$ as well.

When the adversary observes a path P_j ending in an unflushed mix, he must take into account that the probability of this path must reflect all possible choices the

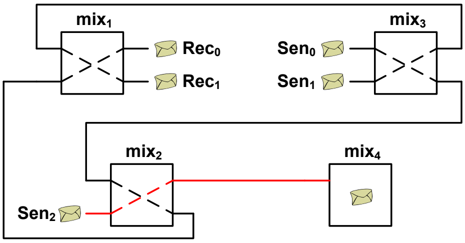


Figure 5.2: Example where the message sent by Sen_2 never arrives to its destination because mix_4 does not receive enough inputs in order to flush (in grey).

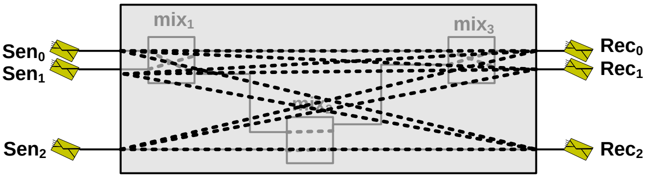


Figure 5.3: Black box abstraction of the system

user could have made. Thus, the attacker computes probability of this path as:

$$\Pr[P_{j,\mathrm{unf}}|\mathcal{C}] = \sum_{l=L_{\mathrm{unf}}}^{L_{\mathrm{max}}} \Pr[L_j = l|\mathcal{C}] \cdot \Pr[\Omega_j|L_j = l, \mathcal{C}] \,.$$

In the above formula, $L_{\mathrm{unf}} = \min(L_{\mathrm{min}}, L_{\mathrm{obs}})$, where L_{obs} is the observed length of the path from the sender until the mix where the message is held. The intuition is that, if the length of the observed unflushed path is larger that the minimum length allowed by the system (L_{min}), the adversary knows the client has not chosen L_{min}, otherwise the message would have already been sent to its receiver. Therefore, the minimum length that could have been chosen is L_{obs}. There is no a priori information in the observation about the maximum length, and the adversary must assume any length up to L_{max} could have been chosen.

As we have done in the previous chapter, we abstract the system as a black box operating as a large threshold mix, reflecting a one-to-one relationship amongst incoming and outgoing messages. (Figure 5.3 depicts an example of this abstraction for the network in Fig 5.1.) In other words, and as indicated in Chapter 3, the messages at the exit of the black box must be a permutation of the messages at the entrance [103]. The number of permutations of N_{msg} messages is $N_{\mathrm{msg}}!$. Without any a priori information, the probability of the real permutation being any of them is: $1/N_{\mathrm{msg}}!$. This information can be easily integrated in the computation of the probability of a set of paths, assuming that users decide independently about the routing of their messages:

$$\Pr[\mathcal{P}|\mathcal{C}] = \left[\prod_{j=1}^{N_{\text{msg}}} \Pr[P_j|\mathcal{C}]\right] \cdot \frac{1}{N_{\text{msg}}!} \,. \tag{5.4}$$

Finally, we recall that the probability of a hidden state is proportional to the probability of all users choosing a set of paths $\mathcal{P}$ (see Eq. 5.2). Hence:

$$\Pr[\mathcal{HS}|\mathcal{O},\mathcal{C}] \propto \Pr[\mathcal{P}|\mathcal{C}] = \left[\prod_{j=1}^{N_{\text{msg}}} \Pr[P_j|\mathcal{C}]\right] \cdot \frac{1}{N_{\text{msg}}!} \,. \tag{5.5}$$

5.3.2 Advanced constraints

In this section we present our modeling of *advanced* constraints which account for additional knowledge of the adversary about the users' behavior. The constraints described here can be selectively combined to refine the probabilistic model of the system, resulting in more accurate attacks. We note that our choice of advanced constraints is not comprehensive. The goal of this section is to illustrate the flexibility of our model, and show how it can be easily adapted to accommodate new attacks or constraints.

Bridging & mix preferences

Bridging attacks were proposed by Danezis and Syverson in [77]. These attacks exploit the fact that users of a large anonymity network might not know all the routers present in the system. In this case it is possible to "bridge" honest routers considering the knowledge (or ignorance) about subsequent mixes in a path that the originator of the communication has. For example, given a message sent through a honest mix, its path through the network can be "bridged" if either: i) there is only one outgoing mix known by its sender, or ii) if there is only one outgoing mix that is not known by all the senders of the other messages present in the round.

Bridging attacks can be incorporated in our model through the definition of an indicator variable $I_{\text{bridge}}(P_j)$ associated with each path. This variable takes the value 1 if all mixes in a given path P_j are known to the initiator of the path, and is set to 0 otherwise. It guarantees that paths containing nodes unknown to the initiator are assigned probability zero. Using this variable we can incorporate bridging in Eq. 5.3 as follows:

$$\Pr[P_j|\mathcal{C}] = \Pr[L_j = l|\mathcal{C}] \cdot \Pr[\Omega_j|L_j = l, \mathcal{C}] \cdot I_{\text{set}}(P_j) \cdot I_{\text{bridge}}(P_j) \,.$$

This probability can in turn be used in Eq. 5.4 to obtain the probability of a set of paths $\Pr[\mathcal{P}]$.

A probabilistic version of bridging can also be incorporated into the model, moving beyond the possibilistic bridging attacks described in [77]. Detailed knowledge of the attacker as to which client knows which server, as well as their probability of choosing it, can be used to build probability distributions over the paths $\Pr[P_j|\text{Sen}_j, \mathcal{C}]$. Such distributions can represent the knowledge of each sender about the mix network infrastructure, but also any preferences they might have about the choice of mixes. The use of guard nodes [282] in Tor [93] can be modeled in this manner.

Non-compliant clients

Our model so far assumes that all clients make routing decisions according to the standard parameters of the system. This is overwhelmingly the case, since most users will be downloading client software that builds paths for them in a particular and known fashion. We call those clients and the paths they create *compliant.* For example, the Tor [93] standard client will choose paths of length three as well as distinct onion routers. Furthermore the first router will be a “guard” [282] node. However, some users may modify the configuration of their client to chose paths differently.

Paths built by these *non-compliant* clients have different associated probabilities from what our model has assumed so far. We are very liberal with those paths, and make as few assumptions as possible about them. Non-compliant clients may select shorter or longer path lengths than usual in the system, i.e., $L_{\overline{cp}} = L_{\text{min}_{\overline{cp}}}, \ldots, L_{\text{max}_{\overline{cp}}}$ with $L_{\text{min}_{\overline{cp}}} \neq L_{\text{min}}$ and $L_{\text{max}_{\overline{cp}}} \neq L_{\text{max}}$. Furthermore, they may use a multiset of mixes (i.e., a mix can appear more than once in the path) to route their messages. We indicate with $\overline{\mathcal{C}}$ that the path has been constructed by a non-compliant user, and its probability can be computed as:

$$\Pr[P_j|\overline{\mathcal{C}}] = \Pr[L_j = l|\overline{\mathcal{C}}] \cdot \Pr[\Omega_j|L_j = l, \overline{\mathcal{C}}] = \frac{1}{L_{\text{max}_{\overline{cp}}} - L_{\text{min}_{\overline{cp}}} + 1} \cdot \frac{1}{N_{\text{mix}}^l} \,.$$

The first term in the multiplication represents the probability of choosing length l uniformly at random from the interval $[L_{\text{min}_{\overline{cp}}}, L_{\text{max}_{\overline{cp}}}]$. Note that although we have arbitrarily chosen a uniform distribution for the length of the non-compliant paths, the model is flexible enough to accommodate any other distribution instead. The second term is the probability of choosing a sequence Ω_j of l mixes (where a mix can appear several times in the path). Further, the indicator variable $I_{\text{set}}(P_j)$ enforcing the need for selecting distinct nodes on the path has disappeared from the equation with respect to Eq. 5.3.

Bridging attacks are still applicable to non-compliant users, as the fact that they choose routing paths based on their own criterion does not affect the mixes they know. If bridging information is available to the adversary, the indicator $I_{\text{bridge}}(P_j)$ can still be incorporated to the formula to account for user's partial knowledge of the network and increase the accuracy of the attack.

In this work we assume that individual users are non-compliant with probability $p_{\overline{cp}}$. If non-compliant clients are present in the network we calculate the joint probability of all paths assuming that each user is compliant or not independently, and assigning a probability to their path accordingly. We denote P_{cp} and $P_{\overline{cp}}$ the set of paths originated by compliant and non-compliant users respectively. We extend the probability model from Sect. 5.3.1 and derive:

$$\begin{aligned}\Pr[\mathcal{P}|\mathcal{C}] &= \left[\prod_{j=1}^{N_{\text{msg}}} \Pr[P_j|\mathcal{C}]\right] \cdot \frac{1}{N_{\text{msg}}!} \\ &= \left[\prod_{P_i \in P_{\overline{cp}}} p_{\overline{cp}} \Pr(P_i|\overline{\mathcal{C}}) \cdot \prod_{P_j \in P_{cp}} (1 - p_{\overline{cp}}) \Pr(P_j|\mathcal{C})\right] \cdot \frac{1}{N_{\text{msg}}!} \,.\end{aligned}$$

In this formula, the product representing the probability of a set of paths $\mathcal{P} = \{P_j,\, x = 1, \ldots, N_{\text{msg}}\}$ being chosen is decomposed in two products, according to the nature (compliant or non-compliant) of the initiators of these paths.

Integrating social network information

A number of attacks, starting by Kesdogan *et al* in [5, 158], and further studied in [60, 70, 76, 161, 183, 270], show that adversaries can sometimes extract profiles of the “friends” of users. These social profiles can then be integrated in the traffic analysis process to narrow down who the receiver of each sent message is [78]. We are initially concerned with incorporating this information into our basic model, as the discussion of how to extract those profiles has already taken place along Chapter 4.

Let us assume that each sender Sen_j can be associated with a sending profile Ψ_j, i.e., a probability distribution where each element $\Psi_j(\text{Rec}_k)$ expresses the probability of sender Sen_j choosing Rec_k as the recipient of a message. When this information is available the probability distribution of the messages at the exit of the black box being a given permutation of the messages at the entrance is not uniform anymore. Hence, Eq. 5.4 does not apply anymore. Instead, we can include the information given by the profiles on the individual paths' probability calculation as follows:

$$\Pr[P_j|\mathcal{C}] = \Pr[L_j = l|\mathcal{C}] \cdot \Pr[\Omega_j|L_j = l, \mathcal{C}] \cdot I_{\text{set}}(P_j) \cdot \Psi_j(\text{Rec}_k)\,,$$

Sen_j being the originator of the path P_j and Rec_k the recipient of her message. This means that the probability of a hidden state would be now proportional to:

$$\Pr[\mathcal{HS}|\mathcal{O},\mathcal{C}] \propto \Pr[P_j|\mathcal{C}] = \Pr[L_j = l|\mathcal{C}] \cdot \Pr[\Omega_j|L_j = l,\mathcal{C}] \cdot I_{\mathrm{set}}(P_j) \cdot \Psi_j(\mathrm{Rec}_k)\,.$$

5.4 A Markov chain Monte Carlo sampler for mix networks

Given an observation $\mathcal{O}$ of some messages' routed through an anonymity network and some knowledge about the constraints $\mathcal{C}$ imposed by its routing algorithms and its users' behavior, traffic analysis aims to uncover the relation between senders and receivers. Equivalently, the goal of traffic analysis is to find the links between incoming and outgoing messages. This comes down to obtaining an a posteriori distribution $\Pr[\mathcal{HS}|\mathcal{O},\mathcal{C}]$ of hidden states $\mathcal{HS}$ given an observation $\mathcal{O}$ and a set of constraints $\mathcal{C}$.

However, enumerating $\Pr[\mathcal{HS}|\mathcal{O},\mathcal{C}]$ for all $\mathcal{HS}$ is computationally unfeasible, due to the very large number of possible hidden states. Instead we have shown in Sect. 5.3 that we can sample states $\mathcal{HS} \sim \Pr[\mathcal{P}|\mathcal{C}]$. These samples are then used to infer the distributions that describe events of interest in the system. For instance, it is easy to estimate the probability $\Pr[i_j \to o_k|\mathcal{O},\mathcal{C}]$ of an incoming message i_j corresponding to any of the outgoing messages o_k as:

$$\Pr[i_j \to o_k|\mathcal{O},\mathcal{C}] \approx \frac{\sum_{\iota \in N_{\mathrm{MH}}} I_{i_j \to o_k}(\mathcal{HS}_\iota)}{N_{\mathrm{MH}}}\,,$$

where $I_{i_j \to o_k}(\mathcal{HS}_\iota)$ is an indicator variable expressing if messages i_j and o_k are linked in hidden state $\mathcal{HS}_\iota$, and N_{MH} is the number of samples $\mathcal{HS} \sim \Pr[\mathcal{P}|\mathcal{C}]$ available to the adversary.

Similarly, we can estimate the sending profile $\Pr[\mathrm{Sen}_j \to \mathrm{Rec}_k|\mathcal{O},\mathcal{C}]$ of a given user Sen_j sending a message to recipient Rec_k. In this case the indicator variable would be $I_{\mathrm{Sen}_j \to \mathrm{Rec}_k}(\mathcal{HS}_\iota)$, indicating whether Sen_j communicates with Rec_k in hidden state $\mathcal{HS}_\iota$:

$$\Pr[\mathrm{Sen}_j \to \mathrm{Rec}_k|\mathcal{O},\mathcal{C}] \approx \frac{\sum_{\iota \in N_{\mathrm{MH}}} I_{\mathrm{Sen}_j \to \mathrm{Rec}_k}(\mathcal{HS}_\iota)}{N_{\mathrm{MH}}}\,.$$

Note that this probability is not the same as $\Pr[i_j \to o_k|\mathcal{O},\mathcal{C}]$ because users can send, or receive, more than one message in an observation [121].

If we compute the probabilities $\Pr[\mathrm{Sen}_j \to \mathrm{Rec}_k|\mathcal{O},\mathcal{C}]$ over all possible k we would obtain an estimation of the sender profile of Sen_j. We note that this profile could

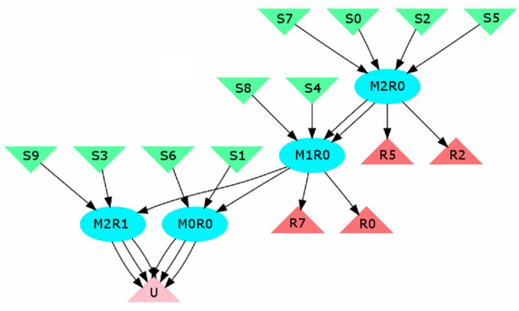


Figure 5.4: Observation of a network where 10 messages are sent to 3 mixes of threshold $t = 4$

be better estimated by integrating the probabilities $\Pr[i_j \to o_k|\mathcal{O}, \mathcal{C}]$ described in this chapter in the Vida model introduced in the Chapter 4, and inferring them in parallel with the de-anonymization process.

We present a Metropolis-Hastings (MH) sampler for $\Pr[\mathcal{P}|\mathcal{C}]$ following the probability mix network model described in Sect. 5.3. For the sake of simplicity in the remainder of the section we omit the conditioning to the observation $\mathcal{O}$ and the constraints $\mathcal{C}$ in all probabilities unless stated differently (e.g., we write $\Pr[i_j \to o_k]$ when we refer to $\Pr[i_j \to o_k|\mathcal{O}, \mathcal{C}]$).

5.4.1 A Metropolis-Hastings sampler for mix networks

Let us consider an anonymity network where users behave as described in Sect. 5.3. An instance of such a network where 10 messages are sent through 3 mixes of threshold $t = 4$ can be seen in Fig. 5.4. This is a simple toy example that we use to illustrate our modeling. We note that increasing the number of messages sent can considerably complicate the observation, see Fig. 5.5. In this figure we still consider that the mixes' threshold is $t = 4$, which is far from reality. Considering more realistic threshold values would likely result in a more complicated observation.

In the figures, senders are represented as triangles and labeled "Sn," n being their identity. Likewise for receivers, represented as triangles labeled "Rn." The triangle labeled as "U" represents *Unknown*, a fake receiver that models the fact that some messages stay in mixes that have not flushed at the end of the observation period. Finally, mixes are represented as ovals, and labeled as "MmRr," where m expresses the identity of the mix and r the round of flushing.

Note that, although we defined the network to consist of three mixes (M0, M1 and M2), in Fig 5.4 messages seem to be sent to 4 different mixes (M0R0, M1R0,

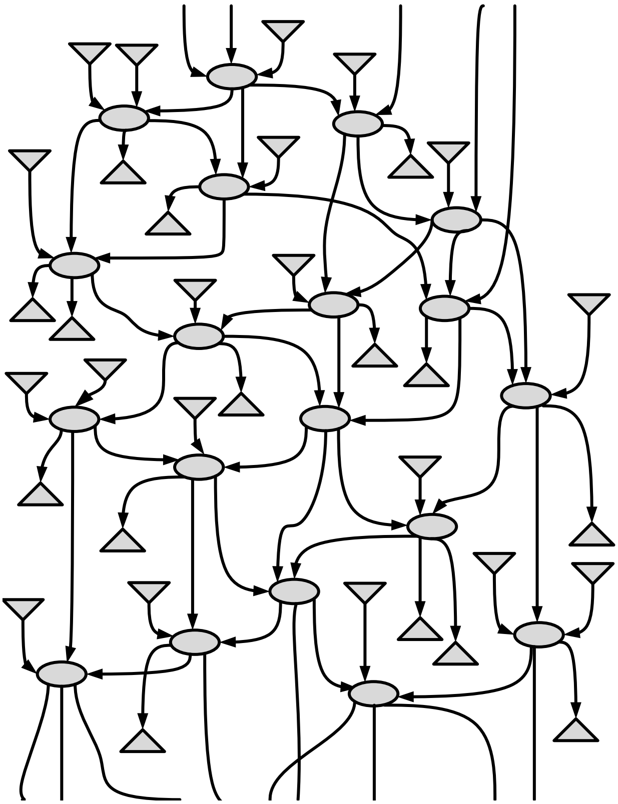


Figure 5.5: Fraction of a larger toy observation

M2R0 and M2R1). This is because of the mixing strategy used by the threshold mix, which empties the memory of the mix after flushing. Thus, messages sent to the same mix in separate rounds do not mix with each other. To illustrate this, let us take mix M2 in our example. Senders S0, S1, S2, S5, S6 and S7 send messages to this mix, that also receives a message output by M1. However, the messages sent by S0, S2, S5 and S7 to mix M2 (M2R0 in Fig 5.4) are flushed before the messages from S1 and S6 arrive to the same mix (M2R1). Therefore the adversary is certain that the outputs of M2R0 do not come from S1 and S6; and the outputs of M2R1 do not (directly) come from S0, S2, S5 or S7, but from S1, S6 or M1.

Let us call the series of mixes that represent a same mix in different rounds as different entities: “virtual mixes;” and denote the set they form as vmixes (in the example vmixes = {M0R0, M1R0, M2R0, M2R1}).

We define a hidden state as a set of internal connections between inputs and outputs in the virtual mixes, such that an input corresponds to one, and only one, output. The aim of the sampler is to provide hidden state samples, according to the actual probability distribution over all possible hidden states. We compute the probability of a hidden state $\Pr[\mathcal{HS}|\mathcal{O},\mathcal{C}] \propto \Pr[\mathcal{P}|\mathcal{C}]$ following the model presented in Sect. 5.3 with both basic and advanced constraints. For simplicity, we denote this probability as $\Pr[\mathcal{HS}]$ in the remainder of the section.

We now explain how to ensure that the random walk performed by the Metropolis-Hastings algorithm actually provides samples from the target distribution $\Pr[\mathcal{HS}]$. In our sampler we select an arbitrary initial state to start the Markov Chain. Given a state $\mathcal{HS}_\iota$ and a transition Q that leads to the candidate state $\mathcal{HS}'$, we decide whether $\mathcal{HS}'$ is a suitable next state for the walk by computing α:

$$\alpha = \frac{\Pr[\mathcal{HS}'] \cdot Q(\mathcal{HS}_\iota|\mathcal{HS}')}{\Pr[\mathcal{HS}_\iota] \cdot Q(\mathcal{HS}'|\mathcal{HS}_\iota)} .$$

The new state $\mathcal{HS}'$ is accepted with probability 1 if $\alpha \geq 1$ or with probability α otherwise, as the Metropolis-Hastings algorithm dictates (Sect. 5.2).

The algorithm requires a proposal probability distribution $Q(\mathcal{HS}'|\mathcal{HS}_\iota)$ according to which candidate states $\mathcal{HS}'$ to continue the random walk are selected. We describe in the next sections a possible proposal strategy for our mix network model. We explain how to obtain $Q(\mathcal{HS}_\iota|\mathcal{HS}')$ and $Q(\mathcal{HS}'|\mathcal{HS}_\iota)$ to be used when computing α.

Basic constraints

When only basic constraints (see Sect. 5.3.1) are considered we define two transitions for the proposal of states:

- $Q_{\mathbf{none}}$: this transition does not change the current state (i.e., the current state is the candidate for next state in the walk),
- $Q_{\mathbf{swap}}$: this transition creates a candidate state by swapping two internal connections in a virtual mix (See Fig. 5.6).

The transition Q_{none}, although seemingly trivial, is necessary to ensure that the Markov Chain resulting from the sampling is ergodic (see Sect. 4.2). To illustrate this need, let us assume the hidden states we are sampling form a fully connected bipartite graph with no self loops. In this graph all transitions take the chain from one set in the bipartite graph to the other. Hence, the initial state and the number of iterations determine in which of the disjoint sets of the bipartite graph the sampler is at each point, biasing the result of the Montecarlo simulation. A priori we have no knowledge about the probability distribution over hidden states,

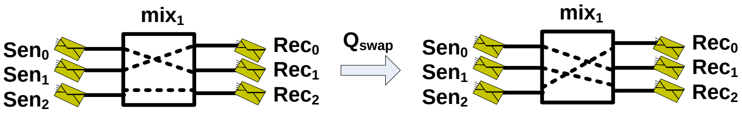


Figure 5.6: Q_{swap} transition operation on the second and third links of a mix

we thus introduce Q_{none} to ensure that even rare cases, as the bipartite graph example, do not affect the result of the sampling.

$Q(\mathcal{HS}'|\mathcal{HS}_\iota)$ (and conversely $Q(\mathcal{HS}_\iota|\mathcal{HS}')$) is the probability of selecting state $\mathcal{HS}'$ as candidate given that the previous state was $\mathcal{HS}_\iota$ (respectively $\mathcal{HS}'$). It depends on the transition Q selected and the probability of selecting this transformation ($\Pr[Q_x]$, $x = \text{none}, \text{swap}$). The values of $\Pr[Q_x]$ are not key for the correctness of the sampling, but have some effect on the mixing speed of the random walk. In our evaluation $\Pr[Q_x]$ is chosen experimentally such that the walk converges fast. Given that a transformation Q_{none} or Q_{swap} has taken place, we compute $Q(\mathcal{HS}'|\mathcal{HS}_\iota)$ as:

$$Q(\mathcal{HS}'|\mathcal{HS}_\iota) = \begin{cases} \Pr[Q_{\text{none}}] \cdot 1 & \text{if } Q_{\text{none}} \\ \Pr[Q_{\text{swap}}] \cdot \frac{1}{|\text{vmixes}|} \cdot \frac{1}{\text{t}} \frac{1}{t-1} & \text{if } Q_{\text{swap}} \end{cases}$$

When the chosen transition is Q_{none}, the candidate state $\mathcal{HS}'$ is the same as the current state with probability 1. If on the other hand the selected transition is Q_{swap}, the probability of choosing a candidate state $\mathcal{HS}'$ is the probability of choosing one of the virtual mixes (vmixes) in the observation and choose two of its links to be swapped.

Advanced constraints: non-compliant clients

When taking into account non-compliant clients, the hidden states are not anymore uniquely defined by the set of internal connections in the virtual mixes "present" in the observation $\mathcal{O}$. In this case a client Sen_j can be compliant or non-compliant ($\text{Sen}_{j,cp}$ or $\text{Sen}_{j,\overline{cp}}$, respectively) resulting in a different probability for the path P_j it initiates, and hence leading to different hidden state probabilities $\Pr[\mathcal{HS}]$. We augment the hidden state to include the internal connections in the virtual mixes, as well as sender labels ($\text{Lab}_j = x$, $x = cp, \overline{cp}$ for sender Sen_j initiator of path P_j) indicating whether we assume that the sender takes routing decisions compliant with the system or not. Further we also define path labels Comp_j that denote whether a path P_j is compliant, i.e. it is built according to the standard parameters of the system ($\text{Comp}_j = cp$); or non-compliant ($\text{Comp}_j = \overline{cp}$).

In this augmented model the random walk Q must modulate the path's labels as compliant or not. Thus, the transitions between states must ensure that senders' labels can change. For this purpose, every time a path is altered by a swap operation, we change its label with probability $p_{\text{flip}}(a,b)$; where $a = \text{Lab}_j$ in $\mathcal{HS}_\iota$, and $b = \text{Comp}_j$ in $\mathcal{HS}_\iota$. At each iteration $\iota+1$ we choose the probability $p_{\text{flip}}(a,b)$ to depend on two factors:

- the label Lab_j that sender Sen_j had in the previous iteration, i.e., in hidden state $\mathcal{HS}_\iota$.
- whether the new path in the candidate state $\mathcal{HS}'$ complies with the system standard parameters or not

Therefore we define four values for $p_{\text{flip}}(a,b)$:

$$p_{\text{flip}}(a,b) = \begin{cases} p_{\text{flip}}(cp,cp) & \text{if } \text{Lab}_j = cp \text{ in } \mathcal{HS}_\iota \text{ and } \text{Comp}_j = cp \text{ in } \mathcal{HS}' \\ p_{\text{flip}}(cp,\overline{cp}) & \text{if } \text{Lab}_j = cp \text{ in } \mathcal{HS}_\iota \text{ and } \text{Comp}_j = \overline{cp} \text{ in } \mathcal{HS}' \\ p_{\text{flip}}(\overline{cp},\overline{cp}) & \text{if } \text{Lab}_j = \overline{cp} \text{ in } \mathcal{HS}_\iota \text{ and } \text{Comp}_j = \overline{cp} \text{ in } \mathcal{HS}' \\ p_{\text{flip}}(\overline{cp},cp) & \text{if } \text{Lab}_j = \overline{cp} \text{ in } \mathcal{HS}_\iota \text{ and } \text{Comp}_j = cp \text{ in } \mathcal{HS}' \end{cases} \tag{5.6}$$

As with other parameters, the actual value of this probabilities affect only the mixing speed of the chain, not its correctness. In our experiments we choose them empirically to ensure fast mixing.

Augmenting the hidden state to include non-compliant senders affects the proposal probability distribution $Q(\mathcal{HS}'|\mathcal{HS}_\iota)$, that now must account for the probability of flipping the labels assigned to senders. In order to integrate this information in $Q(\mathcal{HS}'|\mathcal{HS}_\iota)$ we define an auxiliary variable π_{flip}. Let us assume that in the transformation from $\mathcal{HS}_\iota$ to propose $\mathcal{HS}'$ paths P_x and P_y have been the subject of a link swap, and that $\text{Lab}_{j,\iota}$, $\text{Lab}_{j,'}$, $\text{Comp}_{j,\iota}$, $\text{Comp}_{j,'}$ denote the labels of the senders and the compliance of the paths in both states. Then π_{flip} represents the probability of sender Sen_x being assigned $\text{Lab}_{x,'}$, and sender Sen_y being assigned $\text{Lab}_{y,'}$, in $\mathcal{HS}'$ given that in the current state $\mathcal{HS}_\iota$ they had labels $\text{Lab}_{x,\iota}$ and $\text{Lab}_{y,\iota}$, respectively. We can compute this probability as follows:

$$\pi_{\text{flip}} = \prod_{j=\{x,y\}} p_{\text{flip}(\text{Lab}_{j,\iota},\text{Comp}_{j,'})} \cdot I_{\text{Lab}_{j,\iota} \neq \text{Lab}_{j,'}} + (1 - p_{\text{flip}(\text{Lab}_{j,\iota},\text{Comp}_{j,'})}) \cdot I_{\text{Lab}_{j,\iota} = \text{Lab}_{j,'}} \cdot$$

In this formula $I_{\text{Lab}_{j,\iota}=\text{Lab}_{j,'}}$ and $I_{\text{Lab}_{j,\iota}=\text{Lab}_{j,'}}$ are two indicator variables that indicate whether the label of sender Sen_j is the same in $\mathcal{HS}$ and $\mathcal{HS}'$ or not, respectively. For each of the two paths, we check whether its sender's label has changed to know whether the transition happened with probability $p_{\text{flip}(a,b)}$ or (1-$p_{\text{flip}(a,b)}$), and take the appropriate $p_{\text{flip}(a,b)}$ accordingly to Eq. 5.6.

A further consideration that must be taken while proposing states is that it can be the case that some input messages are assigned to a "deterministic" path. This can happen, for example, when a message immediately enters a mix that is never flushed. Given the proposal strategy we have described so far the unflushed mix would never be eligible for a swap, and the label of this message's sender would remain unchanged throughout the simulation. As a result, some possible hidden states would never be visited by the random walk. In order to ensure that the sampler explores the full state space we define a third type of transition Q_{det}, chosen with probability $\Pr[Q_{\text{det}}]$:

- $\boldsymbol{Q_{\textbf{det}}}$: this transition modifies the compliant status of the sender of one of the N_{det} deterministic paths present in the network. If no clients are deemed to be non-compliant or no deterministic paths exist, this transition is never applied ($\Pr[Q_{\text{det}}] = 0$).

Finally, we integrate π_{flip} and Q_{det} in $Q(\mathcal{HS}'|\mathcal{HS}_\iota)$. Depending on which transition has been selected to propose $\mathcal{HS}'$:

$$Q(\mathcal{HS}'|\mathcal{HS}_\iota) = \begin{cases} \Pr[Q_{\text{none}}] & \text{if } Q_{\text{none}} \\ \Pr[Q_{\text{swap}}] \cdot \frac{1}{\text{vmix}_{max}} \cdot \frac{1}{\text{t}} \frac{1}{t-1} \cdot \pi_{\text{flip}} & \text{if } Q_{\text{swap}} \\ \Pr[Q_{\text{det}}] \cdot \frac{1}{N_{\text{det}}} & \text{if } Q_{\text{det}} \end{cases}$$

Given these three possible transitions: Q_{none}, Q_{swap}, and Q_{det}, our sampler's flow of operations is illustrated in Fig. 5.7. In the diagram u is an auxiliary variable to express the choices made during each iteration.

5.5 Evaluation

The aim of our evaluation is to ensure that the inferences drawn from the Metropolis-Hastings samples are "correct." In the context of this work, correctness means that the a posteriori distributions returned by the sampler represent indeed the probabilities of correspondences between incoming and outgoing messages. In other words, the probabilities that we estimate using the output of the sampler represent the real probability with which events happen in an observation.

We evaluate the inference engine with small (3 mixes) and larger (5 to 10 mixes) networks. For these networks, we create different observations inserting N_{msg} messages ($N_{\text{msg}} \in \{10, 50, 100, 1000\}$) from users that choose paths of length

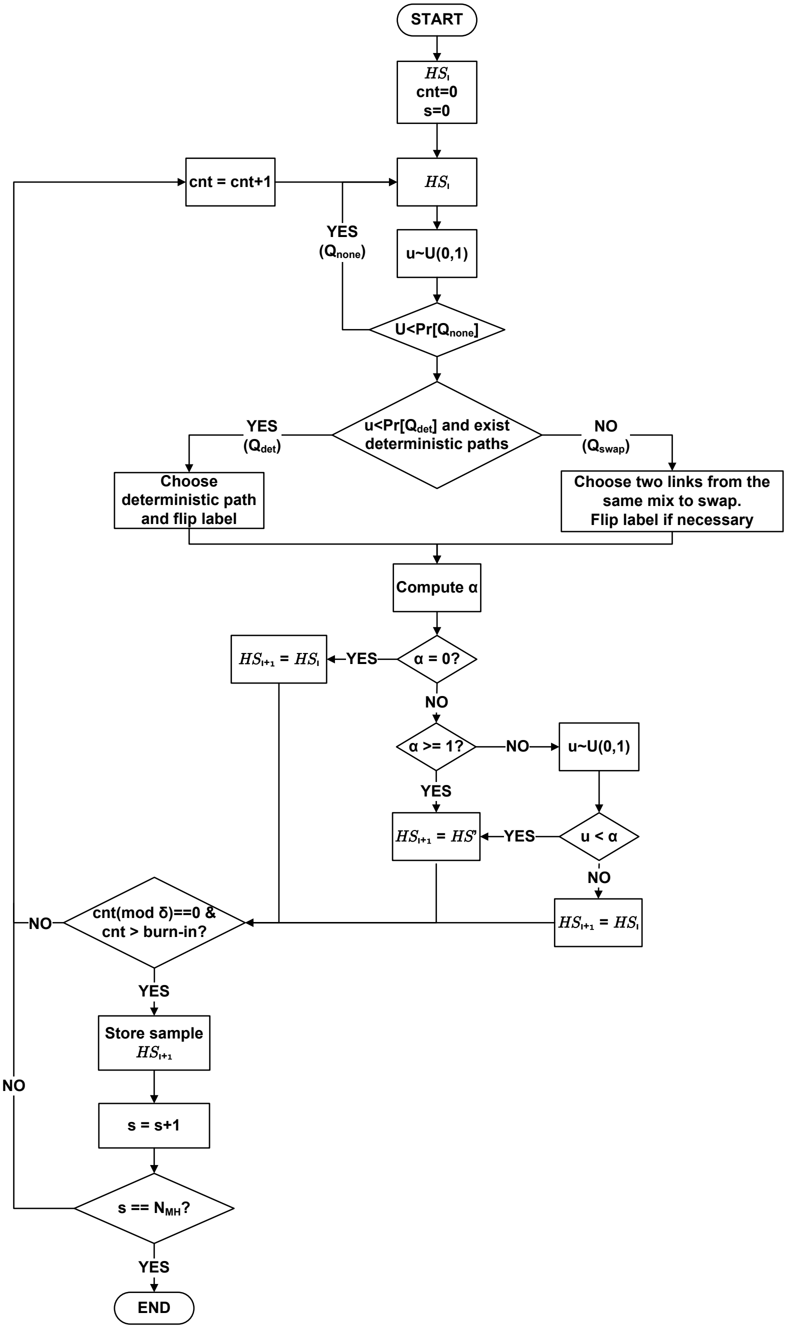


Figure 5.7: Flowchart of our Metropolis-Hastings sampler for mix networks.

between $L_{\min} = 1$ and $L_{\max} = 3$ and select the mixes belonging to these paths uniformly at random. In some of the experiments, we consider the users to be non-compliant with probability $p_{\overline{cp}} = 0.1$. This means that on average there are 10% of non-compliant clients in each observation.

5.5.1 Metropolis-Hastings parameters

The sampler parameters are important to ensure that the samples returned are from the desired distribution $\Pr[\mathcal{HS}]$.

The number of iterations δ the sampler runs between two output samples must guarantee these samples are independent. There is no straightforward procedure to obtain the optimal value for this parameter and we have to estimate it. We consider δ to be large enough when the second order statistics of the marginal distributions $\Pr[i_j \to o_k]$ (respectively $\Pr[\text{Sen}_j \to \text{Rec}_k]$) are the same as the first order statistics. Informally we want to ensure that the probability that an input i_j corresponds to an output o_k at sample ι is independent of the output of i_j at sample $\iota - 1$. Formally, the property we are looking for is:

$$\Pr[i_j \to o_k \text{ in } \mathcal{HS}_\iota | i_j \to o_h \text{ in } \mathcal{HS}_{\iota-1}] = \Pr[i_j \to o_k \text{ in } \mathcal{HS}_\iota] \,, \tag{5.7}$$

for any output o_h. We experimentally test different values of δ to determine a suitable number of iterations that the sampler must run before outputting an independent sample. We note that any higher δ would also ensure independence between samples.

The higher the number of samples N_{MH} extracted, the better the estimate of the a posteriori distributions at the cost of more computation. If we use few samples for our estimation, the estimations regarding events with low probability is likely to have poor quality. In our experiments we choose the number of samples we collect to estimate probabilities based on the order of magnitude of the a posteriori probabilities we expect to infer.

When adapting our experiments to consider non-compliant clients, we need to choose a value for the parameters $p_{\overline{cp}}$, which determines the average percentage of non-compliant clients in the network, and $p_{\text{flip}}(a, b)$, the probability of flipping the senders' labels in a swap operation. We decided to assign $p_{\overline{cp}} = 0.1$ so that the percentage of non-compliant clients using the network is small (as expected in a real network) but their presence in the network has a non-negligible impact on the analysis. We recall that the probability $p_{\text{flip}}(a, b)$ is not crucial for the correctness of the sampler, but is important to the speed of mixing. A study of optimal values for $p_{\text{flip}}(a, b)$ given $p_{\overline{cp}}$[2] is left as subject of future research.

[2] Although in this work we assume $p_{\overline{cp}}$ is known to the attacker, it could be included in the hidden state and inferred together with the rest of hidden variables.

	Parameter	**Value**			
Network parameters	N_{msg}	10	50	100	1000
	N_{mix}	3	3	10	10
	t	3	3	20	20
	$[L_{\min}, L_{\max}]$	[1,3]			
Advanced constraints	$p_{\overline{cp}}$	0.1			
	$p_{\mathrm{flip}}(\overline{cp}, cp)$	0.9			
	$p_{\mathrm{flip}}(\overline{cp}, \overline{cp})$	0.01			
	$p_{\mathrm{flip}}(cp, cp)$	0.02			
	$p_{\mathrm{flip}}(cp, \overline{cp})$	0.3			
	$[L_{\min_{\overline{cp}}}, L_{\max_{\overline{cp}}}]$	[1,32]			
Sampler parameters	δ	6011	6011	7011	7011
	burn-in	8011			
	N_{MH}	500	500	500	500

Table 5.1: Parameters of the Metropolis-Hastings sampler implementation

The values for the parameters used in our experiments are summarized in Table 5.1. We chose the network parameters to produce observations that we can analyze. Had we always considered a realistic mix network, with at least $N_{\mathrm{mix}} = 10$ with threshold $t = 10$, and few messages (10 or 50), we would run the risk of many mixes not flushing and therefore not observing any flow of messages.

5.5.2 Evaluation methodology

For a given observation, we collect N_{MH} samples from $\Pr[\mathcal{P}]$ ($\Pr[\mathcal{P}] \propto \Pr[\mathcal{HS}]$) using the Metropolis-Hastings algorithm with the transitions Q described in Sect 5.3. Using these samples we estimate the marginal probability distributions $\Pr[i_j \to o_k]$ linking input messages to output messages.

Let us call each of the samples obtained in the MH simulation $\mathcal{P}_\iota$, $\iota \in \{1, \ldots, N_{\mathrm{MH}}\}$. The result of our basic experiment is a point estimate of $\Pr[i_j \to o_k]$ for each of the messages i_j entering the network:

$$\Pr[i_j \to o_k] = \frac{\sum_{\iota \in N_{\mathrm{MH}}} I_{i_j \to o_k}(\mathcal{P}_\iota)}{N_{\mathrm{MH}}} . \tag{5.8}$$

Our methodology aims to establish whether these probabilities are correct.

Our test consists of running our basic experiment over 2000 observations. In each of them we select a random input message (i_j) and a random output message (o_k) as targets and we store the tuple:

$$(\Pr[i_j \to o_k], I_{i_j \to o_k}(\text{trace})) .$$

The first element of the tuple is the inferred probability that i_j corresponds to o_k computed with Eq. 5.8 from the samples output by the MH simulation. The second element, $I_{i_j \to o_k}(\text{trace})$, is an indicator variable that takes the value 1 if o_k actually corresponded to i_j when the trace was generated, and 0 otherwise.

Once these tuples are collected, we make a histogram using the first element, $\Pr[i_j \to o_k]$, for the classification of the tuples. Given that $\Pr[i_j \to o_k]$ is a continuous variable we quantify the interval in 30 "bins" of equal size. We denote as $\text{bin}(a,b)$ the histogram bin corresponding to $\Pr[i_j \to o_k] : a \leq \Pr[i_j \to o_k] < b\, a = \sigma * 1/30\, , b = (\sigma * 1/30) + 1/30\, , \sigma = 0, 1, \ldots, 29$, and denote as $\text{Len}(a,b)$ the number elements in that bin. For each of the bins we compute:

$p_{\textbf{sampled}}(a,b)$: which corresponds to the arithmetic mean of the $\Pr[i_j \to o_k]$ belonging to the tuples contained in the bin:

$$p_{\text{sampled}}(a,b) = \frac{\sum_{\Pr[i_j \to o_k] \in \text{bin}(a,b)} \Pr[i_j \to o_k]}{\text{Len}(\text{bin}(a,b))}\, .$$

The value $p_{\text{sampled}}(a,b)$ represents the expected probability for an event given the MH simulation output (Eq. 5.8).

$p_{\textbf{empirical}}(a,b)$: the 95% Bayesian confidence intervals that represents the "actual" probability with which the targeted events happened in the observations. Given how many tuples there are on a bin and the amount of these tuples whose second element is $I_{i_j \to o_k}(\text{trace}) = 1$ we compute this interval using the Beta function:

$$\tau = \sum_{I_{i_j \to o_k} \in \text{bin}(a,b)} I_{i_j \to o_k}(\text{trace}) + 1\, ,$$

$$\upsilon = \text{Len}(\text{bin}(a,b)) - \tau + 2\, ,$$

$$p_{\text{empirical}}(a,b) \sim \text{Beta}(\tau, \upsilon)\, .$$

The beta distribution can be interpreted as the posterior probability of the parameter $\Pr[i_j \to o_k]$ of a binomial distribution, in which success is defined as i_j corresponds with o_k, after observing τ successes (with probability $\Pr[i_j \to o_k]$ of success); and $(\text{Len}(\text{bin}(a,b)) - \tau)$ failures (with probability $(\Pr[i_j \to o_k])$ of failure). In summary, the 95% confidence interval of this distribution indicates likely values of $\Pr[i_j \to o_k]$ that could have generated the observation.

We note that the test could be also carried on using senders and receivers as targets. As demonstrated in [121] there can be a substantial difference between considering just correspondences amongst input and output messages and considering the identities of the sender and receivers of these messages. The difference with respect to the analysis described above is that the tuples stored would be:

$$(\Pr[\mathrm{Sen}_j \rightarrow \mathrm{Rec}_k], I_{\mathrm{Sen}_j \rightarrow \mathrm{Rec}_k}(\mathrm{trace}))\,.$$

The first element is the estimation of the probability that Sen_j has sent a message to Rec_k computed as:

$$\Pr[\mathrm{Sen}_j \rightarrow \mathrm{Rec}_k] = \frac{\sum_{\iota \in N_{\mathrm{MH}}} I_{\mathrm{Sen}_j \rightarrow \mathrm{Rec}_k}(\mathcal{HS}_\iota)}{N_{\mathrm{MH}}}\,.$$

The second element, $I_{\mathrm{Sen}_j \rightarrow \mathrm{Rec}_k}(\mathrm{trace})$, is an indicator variable that takes the value 1 if Rec_k actually received a message from Sen_j when the trace was generated, and 0 otherwise.

In our experiments we expect the mean $p_{\mathrm{sampled}}(a, b)$ to fall within the interval $p_{\mathrm{empirical}}(a, b)$, i.e. the estimated probability being close to the probability with which events actually happen in the generation of the traces. If this is the case we conclude that the implementation of the Metropolis-Hastings sampler is correct. The size of the confidence interval is also meaningful: small intervals indicate that many samples have been used thus, it accurately represents $p_{\mathrm{empirical}}(a, b)$. On the other hand, if few samples are used to compute the interval (if a bin contains few events), we obtain a poor estimate of $p_{\mathrm{empirical}}(a, b)$ and the results based on it are rather meaningless.

Evaluation methodology example

Let us illustrate the evaluation method with a toy example, in which we observe 5 networks ($\mathrm{Net}_1, \ldots, \mathrm{Net}_5$) from which we collect $N_{\mathrm{MH}} = 5$ samples: $\mathcal{P}_1, \ldots, \mathcal{P}_5$. For simplicity we limit the explanation to the computation $p_{\mathrm{sampled}}(a, b)$ and $p_{\mathrm{empirical}}(a, b)$ for one bin, for instance, $\mathrm{bin}(0.4, 0.433)$. Hence, we only consider events $i_j \rightarrow o_k$ with probability $(0.4 \leq \Pr[i_j \rightarrow o_k] < 0.433)$ through our example. In the first network, for the target event $i_j \rightarrow o_k$, we obtain the following:

$$\begin{gathered} I_{i_j \rightarrow o_k}(\mathcal{P}_1) = 0, \quad I_{i_j \rightarrow o_k}(\mathcal{P}_2) = 1, \quad I_{i_j \rightarrow o_k}(\mathcal{P}_3) = 0, \\ I_{i_j \rightarrow o_k}(\mathcal{P}_4) = 0, \quad I_{i_j \rightarrow o_k}(\mathcal{P}_5) = 1 \ \ . \end{gathered}$$

This means that input message i_j was assigned to o_k in the sets of paths $\mathcal{P}_2$ and $\mathcal{P}_5$, but not in $\mathcal{P}_1$, $\mathcal{P}_3$, and $\mathcal{P}_4$. With this information we can compute:

$$\Pr[i_j \rightarrow o_k] = \frac{\sum_{\iota \in N_{\mathrm{MH}}} I_{i_j \rightarrow o_k}(\mathcal{P}_\iota)}{N_{\mathrm{MH}}} = \frac{0+1+0+0+1}{5} = 0.4\,.$$

Additionally, we note down whether in the generation of the network i_j and o_k are actually the same message. Let us assume that for this first network this was not the case, hence we record $I_{i_j \to o_k}(\text{Net}_1) = 0$.

We run the MH sampler for the other four network instances and at the end of the process we have collected the following tuples in the bin of interest, $\text{bin}(0.4, 0.433)$:

$\text{bin}(0.4, 0.433)$	
Network	$(\Pr[i_j \to o_k], I_{i_j \to o_k}(\text{Net}))$
Net_1	$(0.4, 0)$
Net_2	$(0.4, 0)$
Net_3	$(0.4, 1)$
Net_4	$(0.4, 1)$
Net_5	$(0.4, 0)$

$\text{Len}(\text{bin}(a, b)) = 5$ tuples

With this information, we can compute $p_{\text{sampled}}(0.4, 0.433)$ and $p_{\text{empirical}}(0.4, 0.433)$:

$$p_{\text{sampled}}(0.4, 0.433) = \frac{\sum_{\Pr[i_j \to o_k] \in \text{bin}(0.4,0.433)} \Pr[i_j \to o_k]}{\text{Len}(\text{bin}(0.4, 0.433))} = \frac{5 \cdot 0.4}{5} = 0.4 \,.$$

There are 2 observations in which the experiment "succeeds" (networks Net_3 and Net_4) and 3 in which it fails we know that the actual probability of events $p_{\text{empirical}}(0.4, 0.433)$ is distributed according to:

$$p_{\text{empirical}}(0.4, 0.433) \sim \text{Beta}(2 + 1, 3 + 1) \,.$$

The 95% confidence interval of $p_{\text{empirical}}(0.4, 0.433)$ is $[0.0005, 1]$. As only 5 samples are available, the interval is large and not much confidence can be as to which was the actual probability of the observed events. Thus, even though $p_{\text{sampled}}(0.4, 0.433) = 0.4$ falls in this interval, we cannot have much confidence in the correctness of the sampler. If we would like to increase our confidence on the correctness of these samples, it suffices with collecting more samples such that we have more certainty as to which is the actual probability $p_{\text{empirical}}(0.4, 0.433)$.

5.5.3 Evaluation results

We conducted several experiments considering both the basic constraints and the full model (including non-compliant clients) in small and large networks.

Figure 5.8 shows the result of our evaluation using only basic constraints to generate the trace and model it. The lower graph is a histogram of the number of

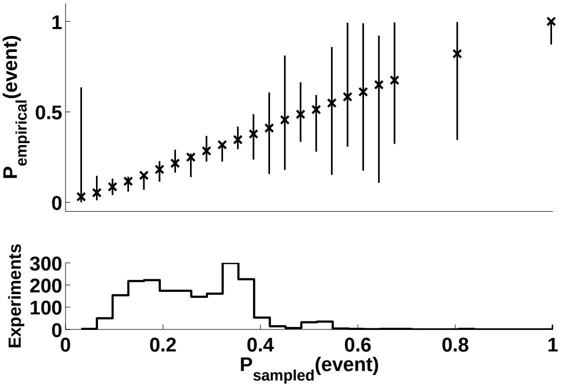


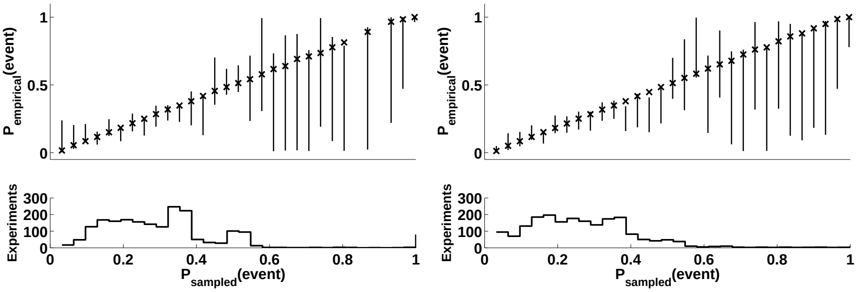
Figure 5.8: Results for the evaluation of an observation generated by 50 messages in a network with $N_{\text{mix}} = 3$ and $t = 3$, when all clients behave in a compliant way

experiments per bin, Len(bin(a, b)). The upper graph represents with crosses the mean of the bins $p_{\text{sampled}}(a, b)$, and the Bayesian confidence intervals $p_{\text{empirical}}(a, b)$ with vertical lines. Most crosses fall in the intervals, meaning that our algorithm is providing samples $\mathcal{HS}_\iota$ according to the correct distribution (only 95% are expected to fall within the intervals). Most messages fall in bins with $p_{\text{sampled}} \in [0.07, 0.4]$, and their confidence intervals are very small, indicating that we have a high certainty our sampler works correctly in that region.

It is noticeable that some paths fall in the $p_{\text{sampled}} = 1$ bin. This denotes total certainty about the correspondence between an input and an output, with no anonymity provided. These are deterministic paths (explained in Sect. 5.4.1) where the attacker is completely sure that the message i_j corresponds to the potential output message o_k because it is the only message inside a mix.

We also performed experiments in which some of the clients behave in a non-compliant fashion. The result for $N_{\text{msg}} = 10$ messages is shown in Fig. 5.9(a). We observe more events with $p_{\text{sampled}} = 1$ that represent deterministic paths. This increase is due to long non-compliant paths ($L_{j,\overline{cp}} >> L_{\max}$) whose links cannot be swapped to form compliant paths.

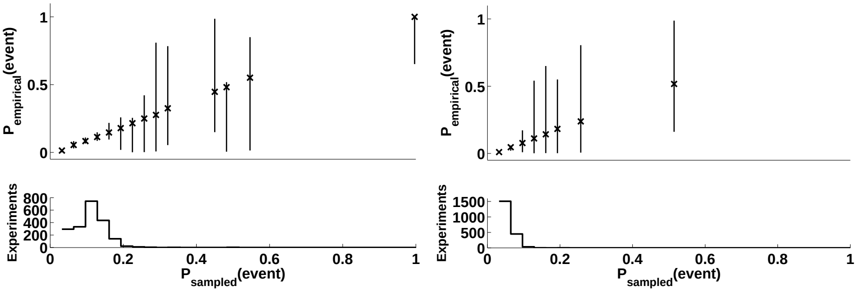
A second difference, with respect to the compliant case, is the appearance of a significant number of events with probability $p_{\text{sampled}} \in [0.7, 1]$. These are paths with no compliant alternative, that now appear as non-compliant paths, with the associated small probability. The probability of these paths is diminished more (generating events with probability $p_{\text{sampled}} \approx 0.7$) or less (generating events with probability $p_{\text{sampled}} \approx 0.95$) depending on how likely the non-compliant path is. These events happen rarely and the number of samples falling in these bins is small, resulting in large confidence intervals.



(a) $N_{\text{msg}} = 10$ messages

(b) $N_{\text{msg}} = 50$ messages

Figure 5.9: Results for the evaluation of an observation of a network with $N_{\text{mix}} = 3$ and $t = 3$, when non-compliant clients are present



(a) $N_{\text{msg}} = 100$, $N_{\text{mix}} = 10$, $t = 20$

(b) $N_{\text{msg}} = 1000$, $N_{\text{mix}} = 10$, $t = 20$

Figure 5.10: Results for the evaluation of big networks

Figure 5.9(b) shows our results when considering 50 messages. As one would expect, we can see in the histogram at the bottom that when more messages travel through the network the attacker is less certain about their destination. There are also fewer samples in the $p_{\text{sampled}} = 1$ bin, which reflects the increase in the anonymity that the presence of more traffic in the network provides to its users.

Finally, we tested the effectiveness of our sampler for longer observations (100 and 1000 messages in the network). The results of the experiments are shown in Fig. 5.10. In these cases, we find that the mix network provides good anonymity for all messages. An attacker cannot link incoming and outgoing messages with a probability higher than $p_{\text{sampled}} = 0.4$ when 100 messages have been observed, and $p_{\text{sampled}} = 0.1$ if more messages are seen.

In all examples, we obtain the expected result: approximately 95% of the samples

Table 5.2: Metropolis-Hastings RAM requirements

$N_{\mathbf{mix}}$	t	$N_{\mathbf{msg}}$	**Samples**	**RAM (Mb)**
3	3	10	500	16
3	3	50	500	18
10	20	100	500	19
10	20	1 000	500	24
10	20	10 000	500	125

fall into the confidence intervals. We conclude that our implementation produces samples from the correct a posteriori probability distribution and implements the optimal Bayesian inference an adversary can perform.

5.5.4 Performance evaluation

Our Metropolis-Hastings sampler is composed by 1443 LOC of Python, including the code associated to the evaluation. Our implementation is not optimized for size, memory usage or running time, and an equivalent implementation in C or C++ would outperform it.

The sampler implementation uses a “two-states” strategy for the proposal and acceptance/rejection of candidates $\mathcal{HS}'$. This strategy stores two states $\mathcal{HS}_0$ and $\mathcal{HS}_1$ that are initialized to the same value (the initial state). In order to propose a candidate we apply a transition Q on $\mathcal{HS}_1$, and compute α (considering $\mathcal{HS}_\iota = \mathcal{HS}_0$ and $\mathcal{HS}' = \mathcal{HS}_1$). If the state is to be accepted, we apply the same transformation to $\mathcal{HS}_0$ ($\mathcal{HS}_0 = \mathcal{HS}_1$). If on the contrary there is a rejection, we undo the modification on $\mathcal{HS}_1$ ($\mathcal{HS}_1 = \mathcal{HS}_0$). Then we restart the process with a new transition Q. This strategy apparently doubles the memory requirements, but actually reduces the amount of extra information needed to walk forward and backwards between states, resulting in a smaller total overhead, and significant ease of implementation.

The memory requirements of the sampler are well within the range of a commodity computer. Table 5.2 presents the memory requirements for different sizes of the observation given by the parameters N_{mix}, t, and N_{msg}. More memory is needed as observations $\mathcal{O}$ and consequently samples $\mathcal{HS}$ grow. Furthermore, we keep the samples $\mathcal{HS}_\iota$ in memory, multiplying the overhead for the number of samples collected (double in the case of having 1000 or more messages with respect to the case when only 50 or 10 messages are considered).

Finally, we measured the computation time for processing observations of distinct size. For each of the sizes we collected 100 measurements of the analysis time and averaged over them. These timings are shown in Table 5.3.

Table 5.3: Metropolis-Hastings timings

$N_{\mathbf{mix}}$	t	$N_{\mathbf{msg}}$	δ	**Full analysis** (min)	**One sample** (ms)
3	3	10	6011	4.24	509.12
3	3	50	6011	4.80	576.42
10	20	100	7011	5.34	641.28
10	20	1 000	7011	5.97	716.72

Computation time increases as the observations increase for two reasons. First, more iterations δ are needed to produce independent samples. Second, the timings include the analysis of all messages in the system, that grow with the observation. Although the time necessary to perform the analysis is already practical, it can be reduced considerably through parallelizing several MH simulations for the same observation to get samples $\mathcal{HS}_\iota$ faster.

5.6 Measuring anonymity

A lot of research has been done regarding the evaluation of anonymity systems. Several tools have been proposed to measure the anonymity provided by these systems [52, 81, 103, 263], amongst which the most popular are the metrics based on Shannon entropy [86, 238]. These metrics are computed over the probability distributions associated with random variables representing user's sending profiles, network level profiles (incoming to outgoing messages correspondences), etc. They give a measure of the uncertainty of the attacker about the possible outcome of the random variable under study.

It is important to realism that the methodology presented in this work does not output a probability distribution, but samples that allow us to approximate probabilities of certain events: $\Pr[i_j \to o_k]$, being i_j an incoming message and o_k an outgoing message. However, only events that have been sampled can be estimated, and we cannot assume that not-sampled events have a null probability. After a finite MH simulation there may be events with very small probability which the random walk has not yet visited (or that have been visited but not sampled) but this does not mean that they are impossible to reach. The estimation of probabilities using MH samples introduces an inherent error coming from the normalization over the sampled events, and not all possible ones. Hence, it cannot be considered a proper probability distribution and it is not possible to measure anonymity by directly applying previously proposed metrics. In this section we explain how to use the MH samples to obtain bounds on the anonymity provided by the system.

Let us consider that we want to measure the anonymity provided by the system to a given message i_j. We denote the probability distribution of this message corresponding to any of the possible N_{msg} outgoing message as $\Psi_j = \{\Pr[i_j \to o_k]$, $y = 1, \ldots, N_{\mathrm{msg}}\}$. Following the approach of Serjantov and Danezis [238] we would measure the anonymity for i_j as the Shannon entropy of this probability distribution:

$$H(\Psi_j) = -\sum_k \Pr[i_j \to o_k] \cdot \log \Pr[i_j \to o_k]\,,$$

but as we said we do not have the full probability distribution, and only samples coming from it.

An approach to the estimation of $H(\Psi_j)$ is to model Ψ_j as a multinomial distribution that determines the probability of outputs o_k corresponding to an input i_j, and resort again to Bayesian inference to estimate it from the samples. For this purpose we also define an auxiliary function that counts the number of times a message i_j is assigned to a message o_k in the set of samples, and denote it as $\mathrm{Ct}_{\mathcal{O}}(i_j \to o_k)$. We note that the Dirichlet distribution is a conjugate prior for the multinomial distribution. A sample from this distribution expresses the belief that the probability of the events $i_j \to o_k$ is $\Pr[i_j \to o_k]$ given that we have observed $\mathrm{Ct}_{\mathcal{O}}(i_j \to o_k)$ occurrences of each of them. Hence we can use the Dirichlet distribution assuming poor prior knowledge over the actual correspondence (Dirichlet(1,...,1)) to obtain samples from Ψ_j [180]. We compute the entropy $H(\Psi_j)$ of n samples Ψ_j from the posterior distribution:

$$H(\Psi_j) \text{ where } \Psi_j \sim \mathrm{Dirichlet}(\mathrm{Ct}_{\mathcal{O}}(i_j \to o_0) + 1, \ldots, \mathrm{Ct}_{\mathcal{O}}(i_j \to o_{N_{\mathrm{msg}}}) + 1)\,.$$

We note that, for the receivers $o_{\overline{k}}$ that do not appear in the samples, $\mathrm{Ct}_{i_j \to o_{\overline{k}}} = 0$.

We order the samples $H(\Psi_j)$ in decreasing order and take as bounds for the anonymity offered by the system the $\gamma\%$ confidence interval for this distribution, i.e., an interval within the range $[0, 1]$, encompassing $\gamma\%$ of the probability mass of the a posteriori distribution.

5.7 Conclusions

In this chapter we have dealt with the computation of probability distributions over correspondences between inputs and outputs of a mix-based anonymity system. Our work has demonstrated that we can extract accurate a posteriori distributions about who is talking to whom, from a complex anonymity system, with a vast hidden state-space, and a large observation. For the first time we are able to calculate the distributions necessary to apply any information theoretic or decision theoretic anonymity metrics.

Our models of mix networks are far from arbitrary: the parameters and architectures we use are inspired by the routing constraints of the deployed mixmaster and mixminion remailers [72]. They can be used to assign to messages a correct degree of anonymity, using probabilistic measures [52, 81, 86, 238, 263], which was not possible before. However, each proposed mix system is slightly different from others, and our model has to still be extended to deal with different mixing strategies [207,240], dummy traffic [75,85,207] as well as observations that start while the mix network is running.

Our model of mix networks is flexible enough to be the basis of such extensions. This has been demonstrated in our comparison of network topologies for low latency, traffic analysis resistant networks [84]. However, it must be stressed that performing efficient inference to estimate the probability of the hidden state in each of these networks might require some craftsmanship.

The traffic analysis methodology we have employed, that defines a probabilistic model over the full system, and performs Bayesian inference to measure the security of the system, is a strong candidate to define the standard by which candidate anonymity systems are proposed and evaluated. In particular the ability to integrate all information in a traffic analysis, as well as extracting probabilities of error, should be seen as essential for proposing robust attacks.

Part II

Design of privacy-preserving systems

Chapter 6

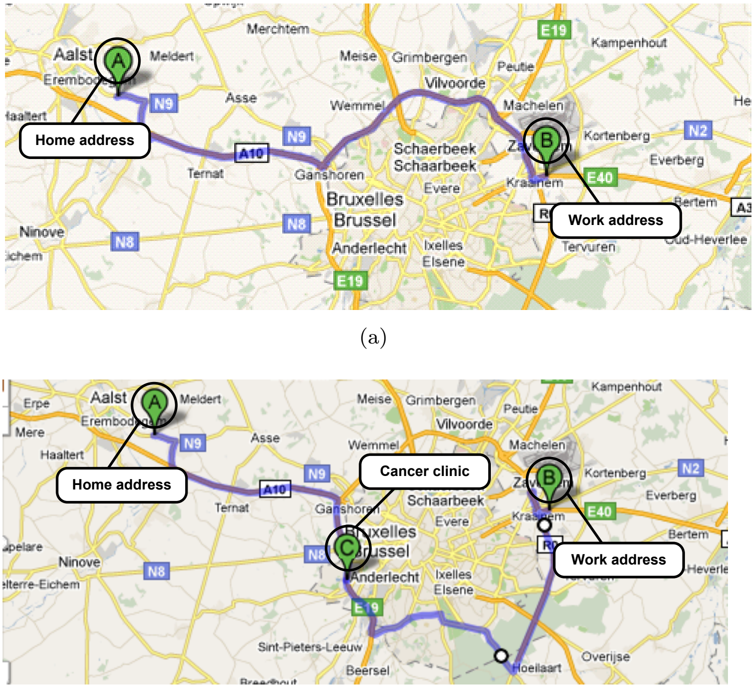
Location privacy: an overview

6.1 Introduction

The widespread of smart mobile devices has fostered the development of a variety of successful location-based services. In these services users share location information in peer-to-peer wireless networks [3, 102, 106], or send their location data to a service provider [4, 105, 149, 235]. In exchange, users enjoy services that may, for instance, enhance their social experience, e.g., a user can look for a perfect dating match in her surroundings [102, 106] or can be able to track one's friends movements in real time [128]; ease their daily activities, e.g., a user can request information about traffic conditions, nearest place of interests (restaurant, gas station, etc.) [216, 271]; or improve their safety in the road, e.g., users' vehicles can communicate to avoid collisions in highways or intersections [33, 287].

Even though location-based services have an enormous potential to benefit service providers and users, these advantages come at a cost for the users. Pervasive communication implicitly generates a large amount of sensitive information encoded in the location and timing where and when this communication takes place. The fact that individuals interact with their environment may allow the service provider, or even passive eavesdroppers, to track users' movements. The analysis of location data can expose aspects of users' private lives that may not be apparent at first, and sensitive information can be inferred from it [127, 132, 151, 167].

Let us consider an example in which a user registers to a real time traffic information service using a fake identity to protect her privacy. From Monday to Friday this user sends to the service provider the route from A to B at 8 am, and the route from B to A at 5 pm, as depicted in Fig. 6.1(a). From these data the provider can infer that with high probability the user lives at address A and works at address B, even if the user gave another set of addresses upon registration. Moreover, crossing this information with a public database (e.g., the U.S. census data) the real identity of the user can be recovered [127]. Now assume that one day this user, for which now the provider can infer identity, home, and work addresses (that the user wanted to keep private), starts sending the trajectory shown in Fig. 6.1(a) on Thursdays' evening. Here the user travels from A to B stopping at C for some time, where C is the address of a known cancer clinic. By revealing this information to the service provider, the user may be unintentionally disclosing highly sensitive medical information.



(a)

(b)

Figure 6.1: Inferring sensitive information from location data: toy example. (This image was created using `http://maps.google.com/`.)

It is not the purpose of this thesis to discuss the implications of revealing fine-grained location data to third parties and we refer the reader to [37] for further details on the consequences of violating location privacy.

Protecting users' location privacy, while enabling them to still benefit from location-based services, is a challenging problem. In this chapter we give an overview of the different techniques for location privacy proposed in the literature and their properties. In our survey, we base our classification in the categories introduced by Shokri *et al.* [247]. We note that this categorization is not strict, in the sense that the privacy-preserving schemes introduced could be classified into more than one category. We choose to assign each system to the category that best defines its privacy protection principles, acknowledging that this choice is not unique. Finally, we would like to emphasize that this chapter is not an extensive survey of the literature but aims to introduce the principles used to protect location privacy.

6.2 Anonymizing unlinkable events

An adversary who can trace an individual along several locations may also profile the individual's behavior over time. A family of solutions to this problem tries to break the linkability of subsequent location samples by changing the identity assigned to them (which can be a one-time or a persistent pseudonym).

In a centralized architecture, in which a trusted third party is in charge of the anonymization process, this is mainly implemented by replacing the users' identities with group pseudonyms, or even having no identity [55, 156]. However, only removing the identity may not be enough as spatio-temporal relations can be exploited to link back the anonymous unconnected samples [56, 131, 167]. To illustrate this, imagine that in the example we used in the previous section the user does not send full trajectories but only samples of her location assigned to a one-time pseudonym. On a trip from home to work the samples the user sends are shown in Fig. 6.2. Given that the samples are timestamped, even if they appear to be sent by different users is not difficult to link them as belonging to the same individual. As we have already discussed once the trajectory is recovered the adversary can infer further information.

Therefore, in addition to identity, it is necessary to protect timing patterns that could be exploited to recover trajectories. Several approaches have been proposed in the literature, all of them relying on the same principle. Inferences are jeopardized by changing the users' identity during a silent period in which the adversary cannot listen to any communications. The assumption is that while no communication takes place the adversary loses track of users and thus their pseudonyms before and after this silent period are unlinkable. This prevents the attacker from recovering full trajectories, reducing the risk of a privacy leakage.

The earliest proposal in this direction are mix zones, by Beresford and Stajano [26], further studied in [41, 113, 114, 116]. *Mix zones* are regions in which users change their credentials while they do not communicate with the environment. When several users traverse a mix zone simultaneously the adversary cannot link ingoing

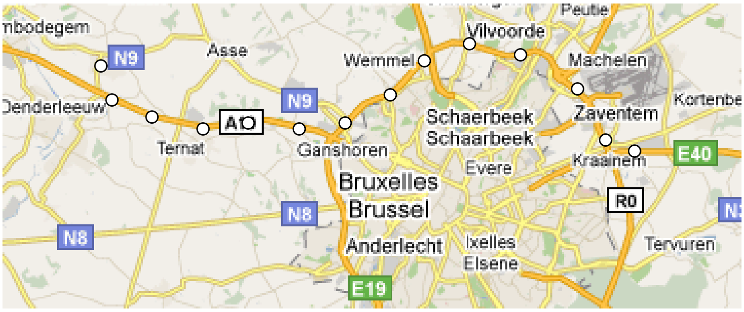


Figure 6.2: Inferring trajectories from location samples: toy example. (This image was created using `http://maps.google.com/`.)

and outgoing users because the new pseudonyms exiting the mix zone could have been chosen by any of the users seen entering the zone. Figure 6.3 illustrates this principle. The adversary observes two users entering the mix zone, and two users leaving. Without prior information both users are equally likely to have chosen Pseudonym 3, respectively Pseudonym 4, as their next identity. Therefore, given the observation on the left the adversary cannot distinguish whether the movements of the users correspond to those in Scenario A or those in Scenario B.

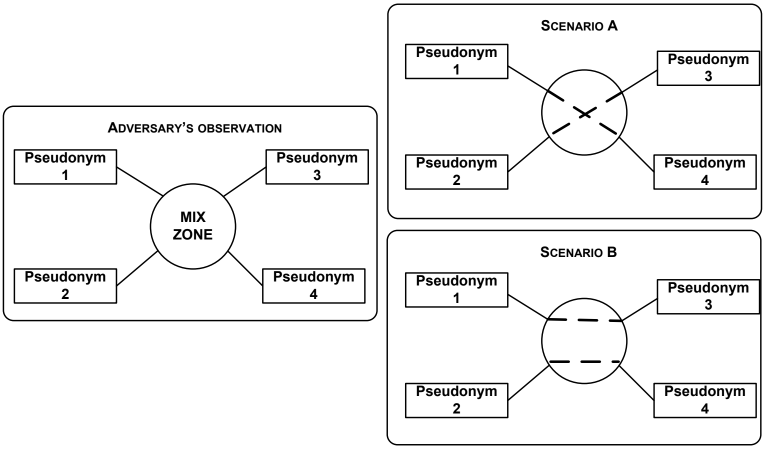


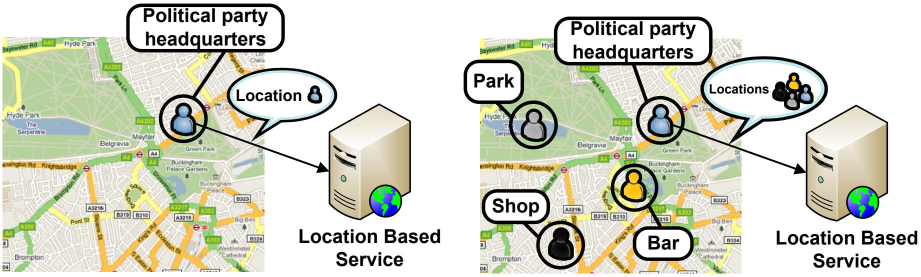
Figure 6.3: Two users traversing a Mix zone. Given the observation the adversary cannot distinguish between Scenario A and Scenario B.

Huang *et al.* [146, 147] follow a similar approach. In their scheme mobile nodes interleave periods of normal communication and periods of silence (no communication at all). During silent periods the nodes' identity is changed in such a way that there is uncertainty of when and where this change takes place.

The silent periods effectively act as mix zones because this uncertainty hinders the adversary's efforts to link samples based on their spatio-temporal relation. The main difference with respect to Beresford and Stajano's system is that when to become silent is an individual choice of the nodes as opposed to a pre-determined location. Random silent periods are also used in AMOEBA [234] for vehicle to vehicle communications. Pre-determined mix zones or random silence periods are not the only options to decide when to change pseudonyms. For instance, Song *et al.* propose to select when to change the identity using the neighboring node density as a threshold [254,255]. The methods can also differ in the cryptographic protocols used for the change of pseudonym (e.g., group signatures [42], or ring signatures [115]).

6.3 Adding dummy events

An alternative to anonymization for achieving location privacy is to add fake samples to the location traces. These *dummy* events, indistinguishable from real actions in the eyes of the adversary, aim to confuse the attacker as to which are the actual movements of the user. Let us assume a user in search for a restaurant that sends her actual position to a location based service provider, as shown in Fig. 6.4(a). The provider may be able to infer the political affiliation from this user. To protect her privacy, the user can send multiple queries corresponding to different locations, as in Fig. 6.4(b), where three out of the four queries correspond to dummy locations. In this case the provider cannot not sure anymore as to which is the actual position of the user preventing easy inferences on users' behavior. Several authors have chosen this approach to mitigate location privacy problems [49,51,163,168,286].



(a) (b)

Figure 6.4: Adding dummy locations to a query: toy example. (This image was created using `http://maps.google.com/`.)

The success of the strategies depends on the ease with which an adversary can tell apart real and fake events. For instance, choosing dummy locations at random may not give good protection as they can fall in the ocean, desert, or in the middle of a lake, hence making filtering a trivial task. Further, real samples have a continuity in time and space which is unlikely to happen with random dummy locations. Duckam *et al.* [99] point out statistical techniques that can be used to filter out false locations in the random walks proposed by Kido *et al.* [163]. The key idea is that random walks do not follow the road network nor have a goal as humans do.

Therefore, care has to be taken when creating plausible false location reports. In order to generate good dummy events for vehicular communications Krumm [168] builds a probabilistic model from GPS tracks from over 250 volunteer drivers. The model accounts for GPS noise, chooses realistic start and end points, plausible driving speeds, etc. A simpler algorithm is introduced by Chow and Golle [51]. They propose to add noise to traces generated by a trip planner. The method is less realistic than Krumm's as people do not always chose optimal routes such as the ones provided by a route planner. Nevertheless the method does not require a database of GPS traces to generate dummy events, which potentially eases its deployment.

Even though many proposals exist, how to generate a trace of events that resembles a normal user's trajectory remains an open problem. See [140] for more details on the difficulty of generating convincing fake data from the point of view of pure statistics.

6.4 Obfuscating events

A third approach to achieve location privacy is to modify the location and/or the timing of events. This adds inaccuracy or imprecision to the adversary's observation [99] hampering inferences on users' behavior. This can be implemented by adding noise to the actual times and or locations, or by coarse graining them.

Among all obfuscation methods, cloaking is by far the most popular protection scheme for location privacy [19, 117, 118, 130, 155, 191, 233, 261, 284, 293]. The concept of *k-anonymity* was originally proposed by Samarati and Sweeney in the field of database privacy [231, 232, 257], where subsets of attributes, called quasi-identifiers, can be used to facilitate the indirect re-identification of individuals in anonymized databases. To overcome this problem, the approach of k-anonymity suggests the suppression and generalization (obfuscation) of quasi-identifiers to make an individual's data entry indistinguishable from others.

In the context of location privacy, the k-anonymity metric was initially adapted to measure location privacy by Gruteser and Grunwald [130]. In this model, each query sent to the service provider (including the user's pseudonym, her position and the query time) is equivalent to one entry in a database, and the location-time information in the query serves as the quasi-identifier. In order to protect a user's location privacy using k-anonymity, each of her queries must be indistinguishable from those of at least $k-1$ other users. To this end, first, the pseudonyms of these k users are removed from their queries. Next, the location-time pair in their queries is obfuscated to the same location-area and time-window, named cloaking region, large enough to contain the users' actual locations. To illustrate the concept let us consider the same example as in the previous section (also shown in Fig. 6.5(a)). In order to protect her privacy, instead of sending her location along with a query, the user can send a region containing three other users (as depicted in Fig 6.5(b)) such that the adversary is uncertain about who is the query issuer.

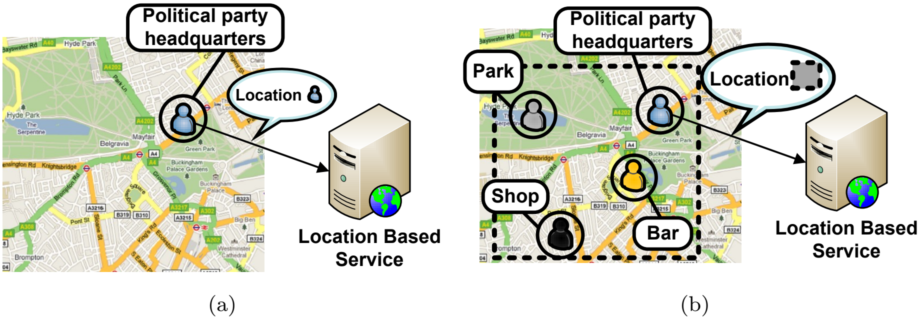


Figure 6.5: Cloaking: a 4-anonymous query. (This image was created using `http://maps.google.com/`.)

The k-anonymity scheme for location privacy has become very popular, mainly due to its simplicity. A large body of research has focused on increasing the efficiency of k-anonymity schemes and reducing the cost of query obfuscation [19, 117, 118, 155, 191, 261, 284], extending the obfuscation method to protect traces [31], or adapting the architecture presented in [130] to different scenarios [233, 293]. Most of these papers take for granted the location-privacy properties enunciated in [130] and focus on improving the quality of service.

In [248] we analyze the effectiveness of k-anonymous cloaking regions for protecting the location privacy of users. We show that there exists a common misunderstanding in the literature, namely a confusion between query anonymity and location privacy. The former refers to the decoupling of a query and the identity of its sender, whereas the latter aims at preventing the adversary from learning the physical location of users. We have shown that by users cloaking a query can be k-anonymous, but their location privacy is not necessarily protected.

Cloaking regions may not be enough to protect location privacy. If the k-anonymous region contains only one point of interest (e.g., a bar, a clinic, etc.) then an attacker can still infer the destination of the user. Bellavista *et al.* [24] propose to create cloaking regions based on Points Of Interest (POI), rather than on other users' positions of the system. The idea is to base the granularity of the region on the number of POIs inside it. The larger the number of POIs the stronger the privacy guarantees as there is more uncertainty as to which POI is the destination of the user.

An alternative scheme was proposed by Ardagna *et al.* [10] in which circular cloaking regions are obfuscated by enlarging the radius, shifting the center, or reducing the radius. This way users can adjust their level of privacy depending on their preferences and the application context.

6.5 Hiding events

There are applications, e.g., traffic monitoring, in which users are forced to disclose their accurate positions over time in exchange for a service. The techniques described in the two last sections are not suitable for these cases. Dummy locations or obfuscated positions would bias the statistics computed at the server defeating the purpose of the application. Further, the accuracy of the released samples enables tracking and re-identification (see Sect. 6.2) discouraging anonymization as a standalone solution to protect privacy.

An alternative is to not only to anonymize the location samples but also remove a subset of them before they are transferred to the server in charge of computing statistics. Two schemes have been proposed that follow this approach, both in a centralized [143, 144] and a distributed [142] architecture. The idea is to remove the location samples in a "clever" manner, such that no trajectory can be recovered from the data. As an example, Hoh *et al.* suggest in [143, 144] to use Shannon's entropy [244] to measure the uncertainty of the adversary when linking location samples. The samples are released only if this uncertainty is above a threshold.

However, hiding some locations is not always desirable. For example in pay-as you-drive applications, such as Electronic Toll Pricing or personalized insurance policies (explained in more detail in the next chapter). In these applications the service provider charges users depending on where and when they drive. For this purpose vehicles carry an On-Board Unit (OBU) that collects the position of the vehicle over time and relays it to the service provider that computes the users' fee [53, 54, 94]. In order to compute this fee the full location record is needed. Therefore, locations cannot be removed from the trace before this computation takes place. A solution to this problem is to allow clients to compute their own fees while hiding the location traces from the provider in such a way that the provider is convinced that the computation was performed correctly.

A number of papers have focused on the design of secure multi-party protocols between the provider and the clients that allow the provider to compute the total fee and detect misbehavior while protecting location privacy. Solutions proposed in [35, 36, 223] resort to general reductions for secure multi-party computation and are very inefficient. A more efficient protocol, VPriv, was proposed in [220]. The dea consists in drivers sending the location data sliced into segments to the provider, in such a way that it is not possible for the latter to link segments belonging to the same client. The provider calculates the subfees of all segments and returns them to all the clients. Each client uses this information to compute her total fee and, without disclosing any location data, proves to the provider that the total fee is computed correctly, i.e., only using the subfees that correspond to the location data input by this particular client. Moreover, in order to prevent malicious users from spoofing the GPS signal to simulate cheaper trips, VPriv has an out-of-band enforcement mechanism. This mechanism is based on the use of random spot checks that demonstrate that a vehicle has been at a location at a time (e.g., a photograph taken by a road-side radar). Given this proof, the provider challenges the client to prove that its fee calculation includes the location where the vehicle was spotted.

The protocol proposed in [220] has several practical drawbacks. First, to provide unlinkability of segments it requires vehicles to send anonymous messages to the server (e.g., by using Tor [93]) imposing high additional costs to the system. Second, their protocol only avoids leaking any additional information beyond what can be deduced from the anonymized database. As the database contains path segments, the provider could use tracking algorithms to recover paths followed by the drivers [131, 143, 167] and infer further information about them. Third, the scalability of the system is limited by the complexity of the protocol on the client side, as it depends on the number of drivers in the system. Practical implementations require simplifications such as partitioning the set of vehicles into smaller groups, thus reducing the anonymity set of the drivers. Fourth, VPriv only uses spot checks to verify correctness of the location, and thus needs an extra protocol to verify the correct pricing of segments. This extra protocol produces an overhead both in terms of computation and communication complexity. Finally, users are required to have a device to carry out the client-side application (e.g., a smart phone or a home personal computer) and a mechanism to transfer the location data from the OBU to this device is needed.

In the next chapter we present a solution, PrETP, that does not require messages between the client and the provider to be anonymous as the computation of the fee is made locally and no location data is sent to the provider. Thus, no database of location data is created and we do not need to rely on database anonymization techniques to ensure users' privacy. Further, the client's operations depend only on the data it collects, independently of the number of vehicles in the system. Contrary to VPriv we leverage the information in spot checks to not only check that clients employ correct location data but also to check that the price paid is correct according to these data, eliminating the need for an extra protocol that makes this verification. Finally, our protocol can be integrated into a stand-alone device without the need of external devices to carry out the cryptographic protocols.

A protocol that employs spot checks to verify both correctness of the location and of the fee calculation is due to de Jonge and Jacobs [80]. In this solution, clients commit to segments of location data and its corresponding subfees when reporting the total fee to the provider. They employ hash functions as commitments. Upon being challenged to ratify the information in the spot check, clients must provide the hash pre-image of the corresponding segment, and demonstrate that indeed the location was used to compute the final fee.

The de Jonge and Jacobs' protocol is limited by the fact that using hash-based commitments one cannot prove that the commitments to the subfees add to the total fee. As solution, they propose that the client also commits to the subfees corresponding to bigger time intervals following a tree structure. Each tax period is divided into months, each month is divided into weeks, and so forth, and subfees for each month, week, day,... are calculated and committed. Then, instead of asking the client to open only one commitment containing the instant specified in the proof, the provider asks the client to open all the commitments in the tree that include that instant. This indeed proves that the sum is correct at the cost of revealing much more information to the provider.

PrETP avoids this information leakage. The reason is that, in our scheme, commitments are homomorphic and thus allow the provider to check that the commitments to the subfees add to the total fee without additional data. The use of homomorphic commitments was also proposed and briefly sketched in [80]. However, their scheme does not prevent the client from committing to a "negative" price, which would give a malicious client the possibility of reducing the final fee by sending only one wrong commitment. Given the amount of commitments sent, this "negative" commitment has an overwhelming probability of not being detected by the spot checks.

The solutions we have so far described are adequate for applications in which the service depends on the processing of large amounts of data (either at the client, at the service provider, or at both). If the number of computations required is small, e.g., querying a database for the nearest restaurant to the current location of the user instead of computing a fee over the whole vehicle's trajectory, the client can resort to Private Information Retrieval [50] to query the service provider's database without revealing her position. Schemes that follow this approach [120, 137,162,210,292] do not suffer from the privacy vulnerabilities of obfuscation-based and anonymity-based solution, but incur higher communication and computation costs. Other cryptographic techniques, such as oblivious transfer [222], have also been suggested to hide users' positions [166].

Chapter 7

Privacy-friendly pay-as-you-drive applications

7.1 Introduction

Vehicular communications are viewed by governments and industry as a perfect tool to support new services. In recent years we have witnessed the appearance in many countries of applications such as electronic toll collection [53, 54, 94], automated traffic law enforcement [110, 211], commercial location-based systems [235], personalized vehicle insurances [208], etc.

In this chapter we focus on what are known as Pay-As-You-Drive applications. In these applications customers are charged depending on the roads they use and at what time they drive instead of a fixed monthly or yearly fee. That is, a personalized fee is computed on the driving log according to a policy defined by the service provider. This policy lays out the exact fares for driving depending on the type of road, time of day, etc. In particular, we study two very similar pay-as-you-drive applications: Pay-As-You-Drive Insurance (PAYD) and Electronic Toll Pricing (ETP).

Insurance represents a large fraction of the cost of owning a car. In order to lower costs for both owners and insurers, insurance companies have developed pay-as-you-drive (PAYD) schemes. In contrast to the current pay-by-the-year policy, customers are charged depending on their driving habits. In PAYD, the insurance fees applied to each user are fairer than the ones in the pay-by-the-year scheme, as customers are only charged for their actual road usage. Customers can reduce their monthly bill by choosing cheap itineraries or by just not using their car. This, in turn, would make vehicle insurance affordable for lower-income car users (e.g., young people), for people that user their car occasionally, or for people who wish to have a second vehicle. Besides, PAYD policies can be socially beneficial by encouraging responsible driving, for instance, discouraging youngsters from driving at night. Due to all these advantages, PAYD insurance policies are supported by motorist associations like the National Motorist Association [13] and the American Automobile Association [12]; and they are being widely implemented by insurance companies all over the world like Uniqa Group [272] (Austria), Hollard Insurance [150] (South Africa), MAPFRE [182] (Spain) or Aioi [6] (Japan), among others.

A similar concept can be applied to road taxes. Currently, citizens are charged a flat fee. In Electronic Toll Pricing, on the contrary, ad hoc fees are calculated for each citizen, according to the distance covered and the kind of road used, among others. Studies [157, 175, 291] show that ETP brings benefits to citizens and governments. The former pay only for their actual road use, while the latter can improve road mobility by applying "congestion pricing." This strategy assigns prices to roads depending on their traffic density such that driving in congested roads is more expensive. This in turn will encourage users to search for alternative routes (or even avoid using their vehicles) thus reducing congestion. The European Commission, through the European Electronic Toll Service (EETS) decision [54,94] (and also some states in the United States [53]) are currently promoting Electronic Toll Pricing.

In order to charge clients depending on their road usage, location information must be used. For this purpose, in the pay-as-you-drive architectures proposed so far, both for ETP [53, 54, 94] and for PAYD insurance [208, 256], vehicles carry an On-Board Unit (OBU) that collects the position of the vehicle over time (e.g., with a GPS receiver). These data are in turn used to compute the final fee at the end of the billing period. A straightforward implementation of a pay-as-you-drive system is one in which the computation of this fee is performed by the service provider. In this approach the OBU acts as a mere relay that collects location data (and, depending on the policy, other information related to the vehicle) and sends it to a back end server. This server is in charge of processing the data to obtain a final premium, which is then sent to the client.

A centralized design can be advantageous in some ways, nonetheless there is a downside for privacy. In this design it is usually argued that the users' privacy is preserved if their private information is protected from eavesdroppers and communication providers (e.g., by means of encryption). Indeed, access to the information in the messages can be hidden from these entities. Yet, the traffic data available to them (e.g., the location where communication takes place) can be used to infer private information (as we have discussed in Chapter 6). Even if the traces are anonymized, the driver's identity can be inferred from the traces themselves [127, 167].

Further, in a centralized architecture the service provider must be trusted not to abuse the collected data (e.g., Data Protection legislation [95] in Europe). In general, one of the aims of this legislation is to limit the processing of the collected data to the one necessary for the purpose of the service (in this case billing users according to their road usage). However, a malicious provider with access to users' fine grained location data, as continuous GPS collection produces, is left in a privileged position to make inferences of what is considered highly sensitive information about customers as we discussed in the previous chapter. This information is highly valuable for the service provider when it comes to obtain a business advantage, as it can be used to profile users and offer them better services. Further, the centralization of the service results in the database being a single point of failure, opening the door to accidental leaks [177] or insiders' leaks (e.g., US secret documents published by Wikileaks [2]). Finally, the collection of these data introduces potential privacy risks as massive sales of data [264]; or abuse by state agencies [108, 251, 252].

The first contribution in this chapter is PriPAYD, a privacy-friendly scheme for pay-as-you-drive insurance, where the premium is calculated by the OBU, and only the minimum information necessary to bill the client is received by the insurance company. We provide an overview of our architecture, in which well-understood techniques are combined to give assurance to the user that the insurance company does not get more information than necessary, while granting her (or a judge in case of dispute) access to all the data. Our techniques also permit easy policy management and policy enforcement by the insurer.

Local processing of location data gives strong privacy guarantees because no private data is transferred to the provider. Yet, it has a downside. When the provider receives the raw location data, data mining can be used to find anomalies in the traces and combat fraud. This verification becomes more problematic when no location is revealed by the OBU.

The second contribution in this chapter is PrETP, a privacy-preserving ETP system in which, without making impractical assumptions, On-Board Units i) compute the fee locally, and ii) prove to the service provider that they carry out correct computations while revealing the minimum amount of location data. PrETP employs a cryptographic protocol, Optimistic Payment (OP), in which OBUs send commitments [40] to the locations and prices used in the fee computation to the service provider along with the final fee. These commitments do not reveal information on the locations or prices. Moreover, they ensure that drivers cannot claim that they were at any other location, nor used different prices, from the ones used to create the commitments.

In order to check the veracity of the committed values, we rely on the service provider having access to evidence (e.g., a photograph taken by a road-side radar or a toll gate) that a car was at a specific location at a particular time, as previously suggested in [80, 220]. Upon being challenged with this evidence, the OBU must respond with some information proving that the location point where the vehicle was spotted was correctly used in the calculation of the final fee. To this end, it opens the commitment containing this location, thus revealing *only* the location data and the price for the instant specified in the evidence provided by the service provider. This information suffices for the provider to verify that correct input data (location and price) was used to calculate the fee. We suggest further techniques for fraud detection when such evidence is not available, as is the case of a insurance company, in [267].

Before diving into the details of the schemes it is important to delineate our threat model. Our goal is to provide a comparable level of privacy protection comparable to what road users already expect today. We assume that any adversary with extensive physical control of the car will be able to track it (by simply installing their own tracking system). Our objective is to limit casual and/or deliberate surveillance by the service provider or other third parties with limited physical access to the car, as well as preventing the aggregation of vast amounts of location information in centralized databases. Fine-grained location/timing information should be hard to obtain for any third party except the user, who has the right to audit the bill and ensure its fairness. In summary, we are satisfied that no systemic surveillance risk is introduced beyond what is already possible today.

The results presented in this chapter have been extracted from our original articles: *PriPAYD: Privacy Friendly Pay-As-You-Drive Insurance* published at the *Workshop on Privacy in the Electronic Society 2007* [268], its extended version *PriPAYD: Privacy Friendly Pay-As-You-Drive Insurance* published at the *IEEE Transactions on Dependable and Secure Computing* [267], *PrETP: Privacy-Preserving Electronic Toll Pricing* published at the *USENIX Security Symposium 2010* [17], and *Engineering Privacy by Design* published at the *4th International Conference on Computers, Privacy & Data Protection 2011* [133].

Chapter outline

The rest of this chapter is organized as follows: we present PriPAYD and analyze its security in Sect. 7.2. Section 7.3 introduces PrETP, offers a high level description of our scheme and its cryptographic components, and presents our prototype implementation and its evaluation. We discuss some practical issues and summarize the steps taken while designing our schemes in Sect. 7.4. Finally, we conclude in Sect. 7.5.

7.2 PriPAYD: Privacy-friendly pay-as-you-drive insurance

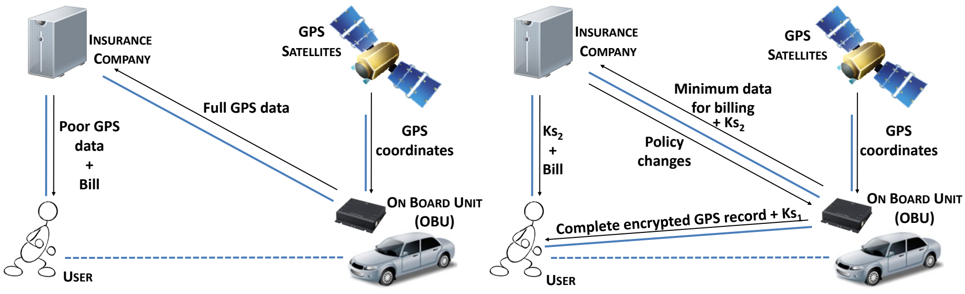
Pay-as-you-drive policies are offered by many insurance companies around the world. This companies gather location data in a variety of ways. We can distinguish three types of policies, based on how privacy-invasive they are. Some of them do not imply any breach of privacy since the amount of kilometers traveled (no location information) needed to compute the premium, is provided only once a year from a fixed location. The second type, despite not recording location information, collects data in geographically distributed points, allowing the insurance company to estimate the movements of the vehicle. Finally, the last model collects GPS data to track all vehicle's location over time. A thorough survey of PAYD implementations belonging to the three categories can be found in [267, 268].

In order to have a reference point against which we can compare $PriPAYD$ we consider the straightforward implementation described in the previous section in which the raw location data is relayed to the provider. This is one of the most privacy-invasive PAYD models that is available today. It works as follows: as the car is being driven, GPS data is collected by the OBU. All these data is sent to the insurance company, who computes the client's premium and send the bill by traditional post, together with a user-friendly summary of the customer's GPS data (see Fig. 7.1(a)). This is very close to the services offered by Octo [208], or Coverbox [57]. The model is a generalization of all the other PAYD approaches, meaning that it can accommodate less privacy-invasive policies (such as those that only take into account yearly odometer readings).

It is important to note that in this model the correctness of the billing depends on the OBU. For this reason, both the customer and the insurer have stakes in its correct functioning, as well as incentives to game it to their advantage. To prevent malicious behavior in practice, the OBUs are provided by the insurance company and should be protected using tamper-evidence and tamper-resistance techniques [8] making it hard for the car user to alter their behavior. Moreover, the car user receives a detailed bill that allows her to audit the vehicle's log and legally challenge the premium if they do not correspond with the actual routes the user had driven.

We present the PriPAYD architecture in Fig. 7.1(b). This architecture follows closely the straightforward implementation, with the exception that the raw and detailed GPS data is never provided to the service provider, or any other third party. The main advantage of PriPAYD, is that the insurance company receives only the billing data, but not the exact vehicle locations (thus cannot infer the users private information) while being sure that the data received is correct. The client can check that only the final premium is being transferred to the insurance company, and the raw data is available for the client to check the correctness of the bill in case of dispute between user and insurer.

Our design safeguards simultaneously the *privacy of the customer* and the *integrity of the billing information.* Yet, similarly to previous PAYD schemes, some attacks against availability cannot be prevented while using cheap, off-the-shelf, technology such as GPS and GSM. Our design attempts to detect that such attacks are taking place, but how they are dealt with has to be the subject of agreement between the insurance company and the customer, and appropriate actions or penalties that deal with them must be codified in the contract. Our guiding design philosophy is that the privacy-friendly mechanisms should introduce no additional vulnerabilities in PAYD with respect to the straightforward implementation.



(a) (b)

Figure 7.1: Straightforward PAYD scheme (a) and Privacy-friendly PAYD model (b).

The key difference between PriPAYD and the straightforward implementation is that the processing of GPS data to obtain the premium data are performed in the OBU. We consider that the data involved in this calculation are the number of kilometers traveled, the hour of the day, the road the user has chosen, and the rate per kilometer depending on the hour and road type (an example policy used by Octo Telematics [208]). To perform the conversion, maps have to be available to the OBU, such that it can match the GPS coordinates with road types. These operations are already supported by any off-the-shelf commercial GPS navigation system or SmartPhone.

The rates imposed by the insurer and other policy parameters can be initialized in the OBU at the time of installation. This information can be updated later in a trustworthy manner through signed updates. For the purposes of this work we consider that policies are uniquely identified by an identifier ID_{policy}. A similar mechanism can be used to perform software upgrades (uploading new firmware to the OBU) with identifier ID_{code}.

Once the premium for a period of time is calculated, the amount to be paid, along with the current policy, ID_{policy}, and code version, ID_{code}, is sent in a secure way to the insurance company. This can be done via GPRS, or even the cheaper SMS services (as currently done by MAPFRE [182], a Spanish insurance company). A timestamp TS is included to prevent reply attacks, in which a client could try to re-submit a message with a small premium later in time. The data is signed by the OBU using a secret (symmetric) key, and encrypted under the public key of the insurance company.

To ensure that the OBU is not acting maliciously in favor of the insurance company, we need to allow a car user or owner to audit the billing. For this purpose, we propose the use of an off-the-shelf USB memory stick. The data is recorded in an encrypted way on this token so that only the customer can access it, and it is signed by the OBU to ensure its authenticity and integrity, and such that it can be used as evidence if there is a dispute. The symmetric encryption key is generated by the OBU and provided to the customer in two shares (that can be used to reconstruct the key): one written on the USB stick and the other relayed through the insurance company and delivered by post with the bill. To ensure forward privacy a mechanism that allows the encryption key to be reset, such as pushing a button on the box for some time, can be put integrated in the OBU. We note that certification is needed to ensure that the box properly resets this key and does not keep old information that may lead to a privacy breach in the future (e.g., when the OBU is returned to the insurance company at the end of the contract). See Sect. 7.4.1 for a more detailed discussion on the certification process.

7.2.1 The security of PriPAYD

At the heart of the PriPAYD security policy we have a two level Bell-La Padula policy [23]: the confidential (high) level contains the sensors and records of the vehicle position and at the restricted (low) level we have the billing information. The only party that is authorized to access the confidential information is the customer, while the insurance company is only authorized to access the billing information. (Note that there is no restriction in the insurance company sending information up to confidential, i.e. policy or software updates.) In this context transferring billing information to the insurance company is an act of *declassification*, since the data at high level is sanitized (only the amount of the final premium is sent) to not leak any information, and sent to low. The provision of the detailed location records by the customer, as part of a dispute, is an even more radical act of declassification.

Three key security properties are required from the channel that transfers the billing data from the vehicle to the insurance company:

Authenticity. Only the OBU can produce billing data that is accepted as genuine by the insurer or any other third party.

Confidentiality. Only the insurer and the car owner should be able to read the billing data transmitted.

Privacy. The customer should be able to verify that *only* the billing data is sent to the insurer.

Authenticity and Confidentiality. A public key signature scheme [186] can be used to certify that the data has been generated and sent by the OBU. As in the straightforward implementation, the signature key in the OBU is difficult to extract due to a custom tamper resistant solution [7] or established smart-card [197] technology. Public key encryption [186] can be used to encrypt the billing information (Data) under the public key of the insurer. There is no key distribution problem since the fingerprints of all public keys are seeded in the box when the device is fitted.

We denote a message sent by the OBU to the insurance company,

$$\mathrm{M} = \mathrm{Enc}_{\mathrm{Insurer\ Key}}(D, \mathrm{Sig}_{\mathrm{Box\ Key}}(D))\,. \tag{7.1}$$

In Eq. 7.1 $D = (\mathrm{Data}, ID_{\mathrm{policy}}, ID_{\mathrm{code}}, TS = \textit{timestamp})$, where ID_{policy} and ID_{code} indicate the policy and the firmware used in the computation of Data. We note that the Privacy property, that allows the user to verify that only billing data is transferred, can also be enforced. Any signature scheme ($\mathrm{Sig}_{\mathrm{Box\ Key}}(\cdot)$) as well as public key encryption scheme ($\mathrm{Enc}_{\mathrm{Insurer\ Key}}(\cdot)$) are verifiable: the customer can be convinced that the encryption is correct by being given the randomness used to perform the encryption operation (in the detailed audit log). The signature can then be verified to ensure it is correctly computed on D. Verifying these only requires the public key of the insurance and the verification key of the OBU, that are public.

Privacy. The task of verifying that no other information is contained in the messages is made difficult by the existence of *subliminal channels* [9, 249] (or covert channels) in signature schemes with the potential to leak information from a maliciously programmed OBU back to the insurance company. Subliminal channels, as well as techniques to limit their capacity, have been extensively studied in the multi-level secure systems literature. PriPAYD implementations should either use signature and encryption schemes that are free from such channels, or estimate their capacity and keep it under a certain threshold [122]. For instance, the client should have control over the source that produced the randomness used in the encryption such that no message can be embedded on it (see Sect. 7.4.1). A further security measure would be to let the user choose when and where does the OBU communicate with the insurance company. This measure avoids covert messages hidden in the time or location where the message was sent and has a positive influence in the privacy-preserving properties of the system (see Sect. 7.4.1). Other ways to give the user full control over the data transmitted would be to use signcryption [164] or a deterministic authenticated encryption scheme [229].

Privacy-friendly auditing. A detailed log of all the vehicle's movements (consisting of location and time) and other audit information can be extracted from the OBU (signed to ensure its authenticity and that the client cannot tamper with the data), by plugging a portable device such as a USB stick on it. However, it should only be accessible to the customer. This is not a trivial requirement to fulfill since the OBU and the customer need to share a symmetric key, unknown to any third party (including the insurance company). We solve the key exchange problem by having the OBU generate the symmetric key and deriving two shares of it (using a secure secret sharing scheme [243], for instance $K_s = K_{s_1} \oplus K_{s_2}$, where $\oplus$ denotes the exclusive or operation). We note that if K_s is not refreshed often enough the amount of data encrypted may jeopardize the security of the system [32]. Thus, we suggest to use a regularly updated session key K' to encrypt the location data, and only encrypt this key under K_s.

It may be the case (e.g. if the insurance and the mechanic collide) that both shares of the key are stolen, in an attempt to compromise the privacy of the customer. To avoid this, any time the OBU is asked to output the encryption key, it creates a fresh pair of shares to be used to encrypt any further data guaranteeing forward security. A user worried that her keys were otherwise compromised can also force the re-initialization of the system. Upon re-initialization the OBU records a fresh key share K_{s_1} on the USB stick, and sends the second fresh share K_{s_2} to the insurance company. To ensure forward secrecy, the old keys and past audit data are securely deleted from the box (see Sect. 7.4.1).

Detection of the OBU's inputs tampering. Even if the insurance company can verify the authenticity of the data and can trust the OBU for correctness, once the box is installed in the car, the company has no control over its environment. A malicious client may try to take advantage of the situation and tamper with the incoming and/or outgoing signals (GPS, GSM, etc.) to reduce the final premium.

Given the difficulty of preventing attacks on technologies such as GSM or GPS, our approach consists in focusing on the detection of such attempts. A solution based on the availability of evidence that a vehicle was at a given location (e.g. photo taken by a road-side radar) is explained in Sect. 7.3. For the sake of brevity in this thesis we omit the description of technical solutions for the cases in which this evidence is not accessible. For further details we refer the reader to [267]. We note that these threats are common for any PAYD model using GPS and GSM technologies, hence the proposed countermeasures should not increase the costs of deploying PriPAYD with respect to the straightforward implementation.

7.3 PrETP: privacy-preserving Electronic Toll Pricing

In the previous section we described a system for pay-as-you-drive insurance in which customers' privacy is guaranteed. To this end, the design choice is to ensure that fine grained location data never leaves the domain of the user. This system strongly relies on tamper resistance for the insurer to believe in the correctness of the computations carried out by the OBU.

In this section we present PrETP, a system based on the same design principles as PriPAYD, that uses cryptographic commitments to prevent fraud. We propose a protocol, Optimistic Payment OP, that makes use of homomorphic commitments which allow the OBU to prove remotely to the service provider that it carries out correct computations, thus relaxing the tampering resistance requirements.

PrETP is optimized for Electronic Toll Pricing, in which we recall citizens are taxed ad hoc fees according to their road usage. The architecture and technologies employed by PrETP are those recommended at European level [54,94], although it could be adapted to other systems, such as [53]. The system model, illustrated in Fig. 7.2 (left), comprises three entities: an On-Board Unit (OBU), a Toll Service Provider (TSP), and a Toll Charger (TC). The OBU is an electronic device installed in vehicles subscribed to an ETP service, and it is in charge of collecting GPS data and calculating the fee at the end of each tax period. The TSP is the entity that offers the ETP service. It is responsible for providing vehicles with OBUs and monitor their performance and integrity. Finally, the TC is the organization (either public or private) that levies tolls for the use of roads and defines what is considered the correct use of the system. In agreement with the TC, the TSP establishes prices for the road usage. Such pricing policy can depend on the type of road (e.g., highways vs. secondary roads), its traffic density, or the time of the day (e.g., rush hours vs. the middle of the night). Additionally, prices can also depend on attributes of the vehicle or of the driver (e.g., low-pollution vehicles, or discounts for retired people).

While the vehicle is driving, the OBU collects the location of the vehicle and calculates the subfees corresponding the trajectories it follows according to the TSP pricing policy. At the end of each tax period, the OBU aggregates all the subfees to obtain a total fee and sends it to the TSP. This process safeguards the

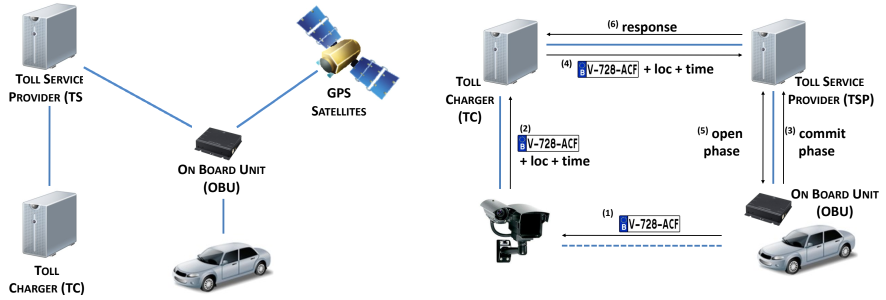


Figure 7.2: Entities in our Electronic Toll Pricing architecture (left). Enforcement spot-check model (right).

privacy of the driver from the TSP, the TC, or any other third party eavesdropping the communications, as no location data leaves the OBU.

Besides preserving users' privacy, the system has to protect the interests of both the TC and the TSP, and provide means to prevent users from committing fraud. Our threat model considers malicious drivers capable of tampering with the internal functionality of the OBU, as well as with any of its interfaces. Under these considerations, we define the security goals of our system as the detection of:

Vehicles with inactive OBUs. Drivers should not be able to shut down their OBUs at will to pretend that they drove less.

OBUs reporting false GPS location data. Drivers should not be able to spoof the GPS signal and simulate a cheaper route than the actual roads on which they are driving.

OBUs using incorrect road prices. Drivers should not be able to assign arbitrary prices to the roads on which they are driving, to lower the final fee. If the policy assigns a price p to a road, drivers cannot use a price $p' < p$ for this road.

OBUs reporting false final fees. Drivers should not be able to report an arbitrary fee, but only the result from the correct calculations in the OBU. If at the end of the tax period the final fee corresponding to the GPS location data collected by the OBU is fee, a driver cannot claim that she must pay $fee' < fee$.

Focusing on the detection of tampering rather that at its prevention allows us to consider a very simple OBU with no trusted components. This has two main advantages: first, reducing the trusted core reduces the production costs of the device. Second, reducing the trusted core to the minimum decreases the number of system components in which the security and privacy guarantees rely, effectively diminishing the risk of security and privacy violations.

In order to perform tamper detection, reliable information about the vehicle's whereabouts is required. We consider that the TC can perform random "spot checks" that are recorded as evidence of the time and location where a vehicle has been seen. Such spot checks can be carried out using an automatic license plate reader, a police control, or even challenging the OBUs using Dedicated Short-Range Communications (DSRC) [54]. Without loss of generality in this work we assume that the evidence is gathered using an automatic license plate reader. This evidence can be used to challenge the OBU to verify its functioning. In order to be able to respond to this challenge while revealing as least location data as possible, the OBU slices the recorded trajectories in segments and computes their corresponding subfees. The sum of these subfees adds up to the final fee transmitted to the TSP. For each segment, the TSP receives a payment tuple that consists of a commitment to location data and time, a homomorphic commitment to the subfee, and a proof that the committed subfee is computed according to the policy. These payment tuples, explained in detail in the next section, bind the reported final fee to the committed values such that the OBU cannot claim having used other locations or prices in its computations. Furthermore, they are signed by the OBU to prevent a malicious TSP from framing an honest driver.

The verification process, depicted in Fig. 7.2 (right), is initiated when the TC gathers evidence about the location of a vehicle at a certain point in time. This information is forwarded to the TSP, along with a request to check that users and OBUs are not misbehaving (i.e., that the security goals enumerated above are met). To this end, the TSP challenges the OBU to open a commitment containing the location and time appearing in the evidence gathered by the TC. The TSP verifies that both challenge and response match, for instance as explained in [220], and reports to the TC whether or the OBU is honest. We assume that the TC (e.g., the government in the EETS architecture) is honest and does not use fake evidence to challenge OBUs.

7.3.1 Optimistic Payment

In this section we sketch the technical concepts necessary to understand the construction of Optimistic Payment, and we outline our efficient implementation of the protocol. For a comprehensive and more formal description of OP, we refer the reader to the original paper [17].

Technical preliminaries

Signature Schemes. A signature scheme consists of the algorithms $\mathsf{SigKeygen}$, $\mathsf{SigSign}$ and $\mathsf{SigVerify}$. $\mathsf{SigKeygen}$ outputs a secret key sk and a public key pk. $\mathsf{SigSign}(sk, x)$ outputs a signature $\mathrm{Sig}_{sk}(x)$ of message x, that we abbreviate as s_x in the reminder of the chapter for the sake of brevity. $\mathsf{SigVerify}(pk, x, s_x)$ outputs accept if s_x is a valid signature of x and reject otherwise. A signature scheme must be correct and unforgeable [126]. Informally speaking, correctness implies that the $\mathsf{SigVerify}$ algorithm always accepts an honestly generated signature. Unforgeability means that no p.p.t. (probabilistic polynomial time) adversary should be able to output a message-signature pair (x, s_x) unless he has previously obtained a signature on x.

Commitment schemes. A non-interactive commitment scheme consists of the algorithms $\mathsf{ComSetup}$, Commit and Open. $\mathsf{ComSetup}(1^k)$ generates the parameters of the commitment scheme $params_{Com}$. $\mathsf{Commit}(params_{Com}, x)$ outputs a commitment c_x to x and auxiliary information $open_x$. A commitment is opened by revealing $(x, open_x)$ and checking whether $\mathsf{Open}(params_{Com}, c_x, x, open_x)$ is true. A commitment scheme has a hiding property and a binding property. Informally speaking, the hiding property ensures that a commitment c_x to x does not reveal any information about x, whereas the binding property ensures that c_x cannot be opened to another value x'. Given two commitments c_{x_1} and c_{x_2} with openings $(x_1, open_{x_1})$ and $(x_2, open_{x_2})$ respectively, the additively homomorphic property ensures that, if $c = c_{x_1} \cdot c_{x_2}$, then $\mathsf{Open}(params_{Com}, c, x_1 + x_2, open_{x_1} + open_{x_2})$.

Proofs of Knowledge. A zero-knowledge proof of knowledge is a two-party protocol between a prover and a verifier. The prover proves to the verifier knowledge of some secret values that fulfill some statement without disclosing the secret values to the verifier. For instance, let x be the secret key of a public key $y = g^x$, and let the prover know (x, g, y), while the verifier only knows (g, y). By means of a proof of knowledge, the prover can convince the verifier that he knows x such that $y = g^x$, without revealing any information about x.

Intuition behind our construction

We consider a setting with the entities presented in the beginning of Sect. 7.3. During each tax period *tag*, the OBU slices the trajectories of the driver in segments formed by a structure containing GPS location data and time. Additionally, this data structure can contain information about any other parameter that influences the price to be paid for driving on the segment. We represent this data structure as a tuple $(loc, time)$. The TSP establishes a function $f : (loc, time) \rightarrow \Upsilon$ that maps every possible tuple $(loc, time)$ to a price $p \in \Upsilon$.

For each segment, the OBU calculates f on input $(loc, time)$ to get a price p, and computes a payment tuple that consists of a randomized hash h on the data structure $(loc, time)$, a homomorphic commitment c_p to its price, and a proof π that the committed price belongs to Υ. The randomization of the hash is needed in order to prevent dictionary attacks to recover $(loc, time)$.

At the end of the tax period, the OBU and the TSP engage in a two-party protocol. The OBU adds the fees of all the segments to obtain a total fee *fee*. The OBU adds all the openings $open_p$ to obtain an opening $open_{fee}$. Next, the OBU composes a payment message m that consists of $(tag, fee, open_{fee})$ and all the payment tuples (h, c_p, π). The OBU signs m and sends both the message m and its signature s_m to the TSP. The TSP verifies the signature and, for each payment tuple, verifies the proof π. Then the TSP, by using the homomorphic property of the commitment scheme, adds the commitments c_p of all the payment tuples to obtain a commitment c'_{fee}, and checks that $(fee, open_{fee})$ is a valid opening for c'_{fee}.

When the TC sends the TSP a proof ϕ that a car was at some position at a given time, the TSP relays ϕ to the OBU. The OBU first verifies that the request is signed by the TC, and then it searches for a payment tuple (h, c_p, π) for which $\mu(\phi, (loc, time))$ outputs accept. Here, $\mu : (\phi, (loc, time)) \rightarrow \{\text{accept}, \text{reject}\}$ is a function established by the TSP that outputs accept when the information in ϕ and in $(loc, time)$ are similar in accordance with some metric, such as the one proposed in [220]. Once the payment tuple is found, the OBU sends the number of the tuple to the TSP together with the preimage $(loc, time)$ of h and the opening $(p, open_p)$ of c_p. The TSP checks that $(p, open_p)$ is the valid opening of c_p, that $(loc, time)$ is the preimage of h and that $\mu(\phi, (loc, time))$ outputs accept.

Intuitively, this protocol ensures the four security properties enunciated in the previous section. Drivers cannot shut down their OBUs, nor report false GPS data as they run the risk of not having committed to a segment containing the $(loc, time)$ in the challenge ϕ. We note that after sending (m, s_m) to the TSP, OBUs cannot claim that they were at any position $(loc', time')$ different from the ones used to compute the message m. Similarly, OBUs cannot use incorrect road prices without being detected, as the TSP can check whether the correct price for a segment $(loc, time)$ was used once the commitments are opened. The homomorphic property ensures that the reported final fee is not arbitrary, but the sum of all the committed subfees. Moreover, by making the OBU prove that the committed prices belong to the image of f, we avoid that a malicious OBU could decrease the final fee by sending only one wrong commitment to a negative price in the payment message, which would give it an overwhelming probability of not being detected by the spot checks. Additionally, the fact that the OBU signs the payment message m ensures that no malicious TSP can frame an OBU by modifying the received commitments, and that a malicious OBU cannot plead innocent by invoking the possibility of being framed by a malicious TSP. Similarly, the fact that the TC signs the challenge ϕ prevents a malicious TSP sending fake proofs to the OBU, e.g. with the aim of learning its location. Finally, the privacy of the drivers is preserved as the OBU does not need to disclose more location information than that in the payment tuple that matches the proof ϕ (already known to TSP).

Efficient instantiation: high level specification

We now outline at high level our efficient instantiation of Optimistic Payment. We employ the integer commitment scheme due to Damgård and Fujisaki [59] and the CL-RSA signature scheme proposed by Camenisch and Lysyanskaya [44]. Both schemes use cryptographic keys based on special RSA modulus n of length l_n. A commitment c_x to a value x is computed as $c_x = {g_0}^x {g_1}^{open_x} \pmod{n}$, where the opening $open_x$ is a random number of length l_n and the bases (g_0, g_1) correspond to the commitment public parameters. Given a public key $pk = (n, R, S, Z)$, a CL-RSA signature has the form (A, e, v), with lengths l_n, l_e, and l_v respectively, such that $Z \equiv A^e R^x S^v (\text{mod } n)$. To prove that a price belongs to Υ, we use a non-interactive proof of possession of a CL-RSA signature on the price. We also employ a collision resistant hash function $H : \{0,1\}^* \to \{0,1\}^{l_c}$.

Initialization. The pricing policy $f : (loc, time) \to \Upsilon$, where each price $p \in \Upsilon$ has associated a valid CL-RSA signature (A, e, v) generated by the TSP, the cryptographic key pair $(pk_{\text{OBU}}, sk_{\text{OBU}})$, the public key of the TSP (n, R, S, Z), the public key of TC, and the public parameters (g_0, g_1) of the commitment scheme are stored on the OBU. Similarly, the TSP possesses its own secret key (sk_{TSP}) and knows all the public keys in the system.

Tax period. Protocol 1 illustrates the calculations and interactions between the OBU and the TSP under normal functioning during the tax period. We denote the operations carried out by the OBU as Pay(), and the operations executed by the TSP as VerifyPayment(). While driving, the OBU collects location data and slices it in segments $(loc, time)$ according to the policy. For each of the N collected segments, the OBU generates a payment tuple $(h_k,\ c_{p_k},\ \pi_k)$. This iterative step is broken down in lines 1 to 21 in Protocol 1. The most resource consuming operation is the computation of π_k, which proves the possession of a valid CL-RSA signature on the price p_k (lines 9 to 20). The length of the random values used in this step is specified in the original paper [17]. At the end of the tax period the OBU generates and signs the payment message m including the tag tag, the total fee, the opening $open_{fee}$, and all the payment tuples $(h_k,\ c_{p_k},\ \pi_k)$, lines 22 to 26. Finally it sends (m, s_m) to the TSP.

Upon reception of a payment message, the TSP executes the VerifyPayment() algorithm. First the TSP verifies the signature s_m using the OBU's public key pk_{OBU}. Next, it proceeds to the verification of the proof π_k included in each of the N payment tuples contained in m, lines 12 to 22. In each iteration it performs

Protocol 1 Protocol between OBU and TSP during taxing phase

```
OBU Pay()

1: // Main loop
2: For all $1 \leq k \leq N$ tuples do:
3:   $p_k = f(loc_k, time_k)$
4:   // Hash computation
5:   $h_k = H((loc_k, time_k))$
6:   // Commitment computation
7:   $open_{p_k} \leftarrow \{0,1\}^{l_n}$
8:   $c_{p_k} = {g_0}^{p_k} {g_1}^{open_{p_k}} \pmod{n}$
9:   // Proof computation
10:    $open_w, w \leftarrow \{0,1\}^{l_n}$
11:    $\tilde{A} = A{g_0}^{w} \pmod{n}$
12:    $c_w = {g_0}^{w} {g_1}^{open_w} \pmod{n}$
13:    $r_\alpha \leftarrow \{0,1\}^{l_\alpha}$
14:    $t_{c_{p_k}} = {g_0}^{r_{p_k}} {g_1}^{r_{open_{p_k}}}$
15:    $t_Z = \tilde{A}^{r_e} R^{r_{p_k}} S^{r_v} (g_0^{-1})^{r_{w \cdot e}}$
16:    $t_{c_w} = g_0^{r_w} g_1^{r_{open_w}}$
17:    $t = c_w^{r_e} (g_0^{-1})^{r_{w \cdot e}} (g_1^{-1})^{r_{open_w \cdot e}}$
18:    $ch = H(\beta || t_{c_{p_k}} || t_Z || t_{c_w} || t)$
19:    $s_\alpha = r_\alpha - ch \cdot \alpha$
20:    $\pi_k = (\tilde{A}, c_w, ch, s_\alpha)$
21: End for
22: // Fee reporting
23: $fee = \sum_{k=1}^{N} p_k$
24: $open_{fee} = \sum_{k=1}^{N} open_{p_k}$
25: $m = [tag, fee, open_{fee}, (h_k, c_{p_k}, \pi_k)_{k=1}^{N}]$
26: $s_m = \mathsf{OBUsign}(sk_{\mathrm{OBU}}, m)$

$\xrightarrow{(m, s_m)}$

TSP VerifyPayment()

$\mathsf{OBUverify}(pk_{\mathrm{OBU}}, m, s_m)$
// Main loop
For all $1 \leq k \leq N$ tuples do:
  $t'_{c_{p_k}} = c_{p_k}^{ch} {g_0}^{s_{p_k}} {g_1}^{s_{open_x}}$
  $t'_Z = Z^{ch} \tilde{A}^{s_e} R^{s_{p_k}} S^{s_v} (1/g_0)^{s_{w \cdot e}}$
  $t'_{c_w} = c_w^{ch} {g_0}^{s_w} {g_1}^{s_{open_w}}$
  $t' = C_w^{s_e} (1/g_0)^{s_{w \cdot e}} (1/g_1)^{s_{open_{w \cdot e}}}$
  $ch' = H(\beta || t'_{c_{p_k}} || t'_Z || t'_{c_w} || t')$
  $ch' \stackrel{?}{=} ch$
  $s_e \in \{0,1\}^{l_e + l_c + l_z}$
  $s_{p_k} \in \{0,1\}^{l_p + l_c + l_z}$
End for
// Commitment validation
$c'_{fee} = \prod_{k=1}^{N} c_{p_k}$
$c_{fee} = {g_0}^{fee} {g_1}^{open_{fee}} \pmod{n}$
$c_{fee} \stackrel{?}{=} c'_{fee}$
```

$$\alpha \in \{p_k, open_{p_k}, e, v, w, open_w, w \cdot e, open_{w \cdot e}\}$$
$$\beta = (n || g_0 || g_1 || \tilde{A} || R || S || g_0^{-1} || g_1^{-1} || c_{p_k} || Z || c_w || 1)$$

a series of modular exponentiations, and uses the intermediate results to compute the hash ch'. Then, it checks whether ch' is the same as the value ch contained in π_k. If this verification, together with the two range proofs in lines 20 and 21, is successful, the TSP is convinced that all the prices p_k used by the OBU are indeed a valid image of f. Finally, the TSP validates the commitments c_{p_k} to ensure that the aggregation of all subfees add up to the final fee (lines 24 to 26). For this, it calculates c'_{fee} as the product of all commitments c_{p_k}, and computes the commitment c_{fee} using the values fee and $open_{fee}$ provided by the OBU. If both values are the same, the TSP is convinced that the final fee reported by the OBU adds up to the sum of all subfees reported in the payment tuples.

Proof Challenge. We denote as $\mathsf{OBUopen}()$ and $\mathsf{Check}()$ the algorithms carried out by the OBU and the TSP, respectively, when the former is challenged with ϕ. When running the $\mathsf{OBUopen}()$ algorithm, the OBU searches for the pre-image $(loc_k, time_k)$ of a hash h_k containing the location and time satisfying ϕ, and sends this information to the service provider along with the price p_k and the opening $open_{p_k}$.

Upon reception of this message, the TSP executes the $\mathsf{Check}()$ algorithm. First, it verifies whether the segment $(loc_k, time_k)$ actually contains the location in ϕ. Then, it computes the value $h'_k = H(loc_k, time_k)$ and checks whether the OBU had committed to this value in one of the payment tuples reported during the tax period. Lastly, the TSP uses $open_{p_k}$ to open the commitment c_{p_k} and verifies whether $p'_k = f(loc_k, time_k)$ equals the price p_k reported by the OBU during the $\mathsf{OBUopen}()$ algorithm. If all verifications succeed, the TSP is convinced that the location data used by the OBU in the fee calculation and the price assigned by the OBU to the segment $(loc_k, time_k)$ are correct.

7.3.2 PrETP evaluation

In this section we evaluate the performance of PrETP. We start by describing the test scenario and both our OBU and TSP prototypes. Next, we analyze the performance of the prototypes for different configuration parameters. Finally, we study the communication overhead in PrETP, and compare it to existing ETP systems.

Test scenario

Policy model. The first step in the implementation of PrETP consists in specifying a policy model in the form of the mapping function $f : (loc, time) \rightarrow \Upsilon$. We decide to follow the same criteria as currently existing ETP schemes [208], i.e., road prices are determined by two parameters: type of road and time of the day. More specifically, we define three categories of roads ('highway', 'primary', and 'others') and three time slots during the day. For each of the possible nine combinations we assign a price per kilometer p and we create a valid signature (A, e, v) using the TSP's secret key. We note that the choice of this policy is arbitrary and that PrETP, as well as OP, can accommodate other price strategies.

Location data. We provide the OBU with a set of location data describing a real trajectory of a vehicle . These data are obtained by driving with our prototype for one hour in an urban area, covering a total distance of 24 kilometers. We note that such dataset is sufficient to validate the performance of PrETP, since results for different driving scenarios (e.g., faster or slower) can easily be extrapolated from the results presented in this section.

Parameters of the instantiation. The performance of OP depends directly on the length of the protocol instantiation parameters, and in particular, on the size of the cryptographic keys of the entities (l_n). In our experiments we consider three case studies: medium security ($l_n = 1024$ bits), high security ($l_n = 1536$ bits), and very high security ($l_n = 2048$ bits). The value l_p is determined by the length of the prices p, which in turn determines the value of l_e. Therefore, both lengths are constant for all security cases. The value of l_v varies depending on the value of l_n. Finally, the rest of parameters (l_h, l_r, l_z, and l_c) are set as the output length of the chosen hash function primitive (see Sect. 7.3.2). These lengths determine the size of the random numbers generated in line 13 in Protocol 1 (see [17] for a detailed explanation). Table 7.1 summarizes the parameter lengths considered for each security level.

Table 7.1: Length of the parameters (in bits)

Parameter	l_n	l_e	l_v	l_p	l_r,l_h,l_z,l_c
Normal Sec.	1 024	128	1 216	32	160
High Sec.	1 536	128	1 728	32	160
Very high Sec.	2 048	128	2 240	32	160

OBU Platform. In order to make our prototype as realistic as possible, we implement PrETP using as starting point the embedded design described in [18], which implements the PriPAYD protocols thus performs the conversion of raw GPS data into a final fee internally. We extend and adapt this prototype with the functionalities of OP to make it compatible with PrETP.

At high-level, the elements of our OBU prototype are: a processing unit, a GPS receiver, a GSM modem, and an external memory module. We use as benchmark the Keil MCB2388 evaluation board [178], which contains an NXP LPC2388 [205] 32-bit ARM7TDMI [11] microcontroller. This microcontroller implements a RISC architecture, it runs at 72 MHz, and it offers 512 Kbytes of on-chip program memory and 98 Kbytes of internal SRAM. As external memory, we use an off-the-shelf 1 GByte SD Card connected to the microcontroller. Finally, we use the Telit GM862-GPS [262] as both GPS receiver and GSM modem.

As our platform does not contain any cryptographic coprocessors, we implement all functionalities exclusively in software. Note that although we could easily add a hardware coprocessor (e.g., [206]) to the prototype in order to carry out the most expensive cryptographic computations, we choose the option that minimizes the production costs of the OBU. Besides, this approach allows us to identify the bottlenecks in the protocol implementation, leaving the door open to hardware-based improvements if needed.

We have constructed a cryptographic library with the primitives required by our instantiation of the OP protocol, namely: i) a modular exponentiation technique, ii) a one-way hash function, and iii) a random number generator. For the first primitive we use the ACL [20] library, a collection of arithmetic and modular routines specially designed for ARM microcontrollers. As hash function we choose RIPEMD-160 [96], with an output length l_h of 160 bits. As our platform does not provide any physical random number generator, we use the Salsa20 [27] stream cipher in keystream mode as third primitive. We note that a commercial OBU should include a source of true randomness.

In order to keep the OBU flexible and easily scalable, we arrange data in different memory areas depending on their lifespan. Long-term parameters ($pk_{\mathrm{OBU}}, sk_{\mathrm{OBU}}, pk_{\mathrm{TSP}}$, commitment parameters) are directly embedded into the microcontroller's program memory, while short-term parameters (payment tuples, $(loc, time)$ segments) and updatable parameters (digital road map, policy f) are stored separately on the SD Card. We note that our library provides a byte-oriented interface with the SD Card, resulting in a considerable overhead when reading/writing values.

TSP Platform. We implement our TSP prototype on a commodity computer equipped with an Intel Core2 Duo E8400 processor at 3 GHz, and 4 Gbyte of RAM. We use C as programming language, and the GMP [111] library for large-integer cryptographic operations.

Performance evaluation

Table 7.2: Execution times (in seconds) for an hour journey of 24 km, for all possible security scenarios.

		Medium Security		**High Security**		**Very high Security**	
Algorithm		**Segment**	**Full trip**	**Segment**	**Full trip**	**Segment**	**Full trip**
Mapping()		76.10 s	839.11 s	76.10 s	839.11 s	76.10 s	839.11 s
Pay()		7.88 s	183.91 s	22.13 s	528.47 s	47.79 s	1 143.30 s
	h_k	0.08 s	1.08 s	0.08 s	1.08 s	0.08 s	1.08 s
	E_k	0.43 s	6.35 s	0.43 s	6.35 s	0.43 s	6.35 s
	c_{p_k}	0.76 s	18.19 s	2.25 s	54.08 s	5.69 s	136.82 s
	π_k	6.20 s	158.09 s	19.45 s	466.96 s	41.64 s	999.05 s

OBU performance. The most time-consuming operations carried out by the OBU during the taxing phase are the Mapping() algorithm and the Pay() algorithm. The Mapping() algorithm is executed every time a new GPS string is available in the microcontroller. Its function is to search in the digital road map the type of road given the GPS coordinates. When the vehicle drives for a kilometer, the OBU maps the segment to the adequate price p_k as specified in the policy. At this point, the Pay() algorithm is executed in order to create the payment tuple. For each segment, the OBU generates: i) a hash value h_k of the location data, ii) a commitment c_{p_k} to the price p_k, and iii) a proof π_k proving that the price p_k is genuinely signed by the TSP (and thus belongs to the image of f). To protect users' privacy we also require that no sensitive data is stored in the SD Card in plaintext form. For this purpose we use the AES [204] block cipher in CCM mode [101] with a key length of 128 bits. We denote this operation as E_k. At the end of the taxing phase, the OBU adds all the prices p_k mapped to each segment to obtain the fee, and all the openings $open_k$ to obtain $open_{fee}$. Finally, the OBU constructs and signs the payment message m and sends it to the TSP.

As it does not involve the key, the computing time of the Mapping() algorithm is independent of the security scenario. Further, this time only depends on the duration of the trip and is independent of the speed of the vehicle: the Mapping() algorithm is always executed 3 600 times per hour, taking a total of 839.11 seconds in our prototype. However, for each of the segments this time can vary depending on the number of points that have to be processed, i.e., depending on the speed of the vehicle. In our experiments it requires 76.10 seconds for the longest segment, i.e., the one where the vehicle spent more time to drive one kilometer and thus $(loc_k, time_k)$ contains the larger number of points. We must stress that the Mapping() algorithm used by our prototype is not optimized for speed, hence the figures we present are an overestimation of the actual times that one could achieve in a commercial implementation.

Similarly, the execution time for h_k and E_k depends exclusively on the length of the segments $(loc_k, time_k)$, as it is proportional to the number of GPS points in the segments. The amount of points per segment varies not only with the average speed of the car but also depending on the length of the segments defined in the pricing policy. In our experiments, computing h_k and E_k takes 0.08 seconds and 0.43 seconds, respectively, for the shortest and the longest segments. For the Mapping() algorithm and both h_k and E_k operations, more than 90% of the time is spent in the communication with the SD card.

On the other hand, the execution time for c_{p_k} and π_k is constant for all segments, as it does not depend on the length of a particular slice (see lines 6 to 20 in Protocol 1). In order to calculate c_{p_k}, the OBU needs to generate a random opening $open_{p_k}$ and perform two modular exponentiations and a modular multiplication. The computation of π_k involves the generation of ten random numbers and a hash value, and the execution of fourteen modular exponentiations, nine modular multiplications, eight additions, and eight multiplications. The bottleneck of both operations is determined by the modular operations. Although we could take advantage of fixed-base modular exponentiation techniques, we choose to use multi-exponentiations algorithms [92], which have less storage requirements. Multi-exponentiation based algorithms, which compute values of the form $a^b c^d (\text{mod } n)$ in one step, allow us to speed up the process. The average execution times for computing c_{p_k} are 0.76 seconds, 2.25 seconds, and 5.69 seconds for medium, high, and very high security respectively. For π_k, these times are 6.20 seconds, 19.45 seconds, and 41.64 seconds, respectively.

Table 7.2 summarizes the timings for all OBU operations and routines for a journey of one hour. We note that, even when 2048-bit RSA keys are used, the OBU can perform all operations needed to create the payment tuples in real time. While the trip lasted one hour, the Mapping() and Pay() algorithms only required 1 982.41 seconds. The computation time is dominated by the Pay() algorithm, which depends on the number of GPS strings in each segment $(loc, time)$. This number varies with the speed of the vehicle and the pricing policy. If a vehicle is driving at a constant speed, policies that establish prices for small distances result in segments containing less GPS points than policies that consider long distances. Similarly, given a policy fixing the size of the segments, driving faster produces segments with less points than driving slower. In both cases, π_k has to be computed fewer times and the Pay() algorithm runs faster. Thus, the policy can be used as tuning parameter to guarantee the real-time operation of the OBU.

Using the values in Table 7.2, for each of the levels of security we can calculate the time our OBU is idle – in our case $(3\,600 - 839.11)$ seconds, with 839.11 seconds being the time required by our non-optimized Mapping() algorithm. Then, considering our current policy, we can estimate the number of times the Pay() algorithm could be executed, which in turn represents the number of kilometers that could have been driven by a car in one hour, i.e., the average speed of the car. For normal security, our OBU could operate in real time even if a vehicle was driving at 350 km/h. This speed decreases to 124 km/h when 1536-bit keys are used, and to 57 km/h if the keys have length 2048 bits. Only when using high security parameters our OBU would have problems to operate in the field. However, as mentioned before, including a cryptographic coprocessor in the platform would suffice to solve this problem whenever high security is required. Also, if the Mapping() algorithm were optimized the OBU would have more time to execute the Pay() algorithm hence being able to handle faster vehicles. Moreover, in our tests we consider a worst-case scenario in which all GPS strings are processed upon reception. In fact, processing fewer strings would suffice to determine the location of the vehicle. As the execution time required by the Mapping() algorithm would decrease linearly, OBUs would be able to support higher vehicle speeds.

In the OBUopen() algorithm, only executed upon request from TC, the OBU searches its memory for a segment $(loc, time)$ in accordance to the proof sent by the TSP. Here, the time accuracy provided by the GPS system is used to ensure synchronization between the data in ϕ and the segment $(loc, time)$. The main bottleneck of this operation is the decryption of the location data corresponding to the correct segment. On average, our prototype can decrypt such a segment in 0.27 seconds.

TSP performance. The most consuming task the TSP must perform corresponds to the VerifyPayment() algorithm, which has to be executed each time the TSP receives a payment message. This algorithm involves three operations: the verification of the proof π_k for each segment, the multiplication of all commitments c_{p_k} to obtain c_{fee}, and the opening of c_{fee} in order to check whether it corresponds to the reported final fee. The most costly operation is the verification of π_k, in particular the calculation of the parameters (t'_{c_m}, t'_Z, t'_{c_w}, t') which requires a total of eleven modular exponentiations (lines 14 to 22 in Protocol 1).

Table 7.3 (left) shows the performance of the VerifyPayment() algorithm for each of the considered security levels when segments have length one kilometer. We also provide an estimation of the time required to process all the proofs sent by OBU during a month, assuming that a vehicle drives an average of 18 000 km per year (1 500 km per month).

Table 7.3: Timings (in seconds) for the execution of VerifyPayment() in TSP (left). Number of OBUs supported by a single TSP (right).

VerifyPayment()	**Segment**	**Month**
Medium Sec.	0.0105 s	15.750 s
High Sec.	0.0295 s	44.250 s
Very high Sec.	0.0587 s	88.050 s

Segment size	**Medium Security**	**High Security**	**Very high Security**
0.5 km	82 000	29 000	14 000
0.75 km	123 000	43 000	22 000
1 km	164 000	58 000	29 000
2 km	329 000	117 000	58 000
3 km	493 000	175 000	88 000

These results allow us to extrapolate the number of OBUs that can be supported by a single TSP in each security scenario for different segment lengths. Intuitively, the capacity of TSP increases when segments are larger, as the payment messages contain fewer proofs π_k. The number of OBUs supported by a single TSP is presented in Table 7.3 (right). For a segment length of 1 km, the TSP is able to support 164 000, 58 000, and 29 000 vehicles depending on the chosen security level. Even when l_n is 2048 bits, only 36 servers are needed to accommodate one million OBUs. This number can be reduced by parallelizing tasks at the server side, or by using fast cryptographic hardware for the modular exponentiations.

Communication overhead

We now compare the communication overhead of PrETP with respect to straightforward ETP implementations and VPriv [220] (see Sect. 6.5). Both in straightforward ETP implementations and in VPriv the OBU sends all the GPS strings to the TSP. Let us consider that vehicles drive 1 500 km per month at an average speed of 80 km/h. For this month, transmitting the full GPS information to the TSP requires 2.05 Mbyte (considering a shortened GPS string of 32 bytes containing only latitude, longitude, date and time). VPriv requires more bandwidth than straightforward ETP systems, as extra communications are necessary to carry out the interactive verification protocol (see Sect. 6.5). Using PrETP, the communication overhead comes from the payment tuples that must be sent along with the fee. For each segment, the OBU sends the payment tuple (h, c_p, π) to the TSP. When sent uncompressed, this implies an overhead of approximately 1.5 Kbyte per segment, i.e., less than 2 Mbyte per month, for medium security (l_n=1024 bits). Additionally, less than 50 Kbyte have to be sent occasionally to respond a verification challenge after a vehicle has been seen at a spot check. As we can see, this overhead is similar to that of the straightforward implementation, although it increases for higher levels of security. We believe that PrETP's communication overhead is not excessive for the additional security and privacy properties the system offers.

The communication overhead in PrETP is dominated by the payment message m sent by the OBU to the TSP. The length of this message depends on the number of segments covered by the driver. Therefore, the segment length can be seen as a parameter that tunes the trade-off between privacy and communication overhead. The smaller the segments, the larger the communication overhead, because more tuples (h_k, c_{p_k}, π_k) need to be sent. Allowing larger segments reduces the communication cost but also reduces privacy because the OBU must disclose a bigger segment when responding a verification challenge.

Further, the communication overhead can be almost eliminated if at the end of each tax period the OBU sends only the hash of the payment message, instead of sending the full sequence of tuples. The downside of this approach is that the TSP loses the ability to remotely check that the fee reported is the sum of the subfees, and that these subfees are computed using genuine prices. Following the spirit of the random "spot checks" used for checking that no GPS spoofing is happening and that correct road prices are used, the OBUs could occasionally be challenged to prove it is operating correctly. To respond this challenge, the OBU would send the payment message corresponding to the preimage of the hash sent at the end of a random tax period. With this payment message the TSP can make the same verifications as in our original description of the protocol.

7.4 Discussion

We have so far provided a technical description of PriPAYD and PrETP. In this section we first discuss some issues related to cost, privacy, certification and practical issues regarding the deployment of our systems. Further discussion on the legal compliance of our systems can be found in [17, 267, 268]. Secondly we revisit the decisions we have made throughout the design of our solution from a methodological point of view, according to the design steps described in [133].

7.4.1 Technical discussion

Cost

In terms of hardware requirements, a processing unit with GSM and GPS interfaces is required for any PAYD or ETP model, hence the only additional hardware required in our system's OBU is an external memory module (e.g., a simple SD card) which should not considerably increase the production costs. Both PriPAYD and PrETP do require more computations and mapping data in the OBU than a straightforward implementation of pay-as-you-drive. Yet, these are comparable to what current commercial GPS navigation systems do. Our OBU prototype, constructed with off-the-shelf components, demonstrates that these systems can be built at a reasonable cost.[1] The additional engineering effort that is required for building a slightly more complex OBU should be more than balanced by the reduced costs of the back end systems, since they handle less, as well as less sensitive, data.

Tamper resistance is needed for the insurance company to trust that the PriPAYD OBU makes correct computations. The security of PrETP's Optimistic Payment scheme does not rely on any countermeasure against physical attacks by drivers. Nevertheless, for liability reasons it is desirable to use OBUs with a certain level of tamper resistance. Since On-Board Units in the market [208, 256] already require tamper resistance, no additional costs should be expected from this either.

Another source of costs is GSM communications. The PriPAYD model should be cheaper since only billing data, instead of raw location data, is sent to the provider. Billing data can be aggregated to further reduce those costs. In the case of PrETP, where more data than in PriPAYD is transferred to the service provider, our analysis in the previous section demonstrates that the overhead with respect to a privacy-invasive scheme is negligible.

Updates for maps and policies can be pushed to the OBU either through the GSM communications or during the servicing of the car. It can be argued that the need for policies and maps updates in PriPAYD an PrETP implies extra communication costs with respect to the straightforward implementation. We note, however that these updates can be considered occasional as it is reasonable to assume that the frequency with which fees are recalculated is low and so is the rhythm with which new roads are constructed and ready for usage thus integrated into maps.

Our architectures keep the trust infrastructure to a minimum, and particularly they do not require a public key infrastructure because there is no need for an external Certification Authority. The identity infrastructure and key management and distribution are based on the pre-existing relationship of the client with the

[1] The cost of our prototype amounts to $500; such a number would be drastically reduced in a mass-production scenario.

insurance company, or of the citizen with the government, respectively. Hence there is no cost associated with either of these.

Strengthening privacy

Some additional privacy concerns should be tackled as part of a real-world implementation.

A first concern is the use of GSM to transfer the data back to the service provider. In our scheme the billing data does not contain any sensitive location information, but an active GSM device registered in the network does leak the cells the vehicle is transmitting from. Hence it is prudent to keep the GSM system powered down at all times except when transmitting. The transmission time and location must be chosen to minimize location leakage because of the GSM technology. Defining and using a preferred known 'home' location, recorded in the box when initialized, should easily address this concern. Still, a timer in the OBU should ensure that, even if the car is not present at this location for a long period of time (e.g. long trip), the monthly premium is sent to the company.

Although our systems protect privacy by keeping the location data in the client domain and exploiting the hiding property of cryptographic commitments, there exist a few sources of information available to the TSP. First, as in many other services, users must subscribe to the service by revealing their identity, and most likely their home address, to the TSP. Second, the final fee and all the commitments must be sent to the TSP at the end of each tax period, and this allows the TSP to estimate the number of kilometers driven. In fact, in our example policy where prices are assigned to entire kilometers, the number of tuples in the payment message leaks the exact distance traveled by the user. The TSP can apply decoding techniques (e.g., [66]) to these data, and infer the trajectories followed by a vehicle by inspecting the possible combination of prices per kilometers that could have generated the total fee. A possible solution to this problem is to give users the possibility to send data associated to dummy segments. In order to keep the correctness of the final fee, we assign a price zero to these dummy segments. Further, we include a price p zero in the pricing policy so that the proofs π_k are still accepted by the TSP. The downside of this approach is that it introduces an overhead in both the computation and communication cost of the system.

Finally, in some cases it is desirable to securely delete past location data to safeguard privacy. For this purpose we advise implementers to never automatically store encrypted GPS data from the audit record; and users to keep this, or key material, only on the USB stick to which they were written by the OBU. This allows the user to easily destroy the data by destroying or deleting the USB stick. Further, the OBU would need a mechanism to reset the encryption key such that no GPS data is encrypted under the destroyed key nor the destroyed key can be recovered. This resetting mechanism needs to be intuitive for the user but such that it cannot be inadvertently activated, e.g., pushing a button on the box for some time, ask for confirmation or PIN code before resetting, etc. We must stress that once audit records of the detailed locations have been deleted it is difficult to challenge any bills that seem incorrect. Hence, the user must be very careful when deciding whether her privacy needs justify her inability to contest the bill.

Certification and independent monitoring

A key objective in our design is to reduce to the minimum the tamper resistance requirements of the OBU to guarantee user's privacy. However, as the OBU is commissioned by the service provider the user has limited capacity to check whether it is functioning correctly or leaking private information. Our design choice is to allow users to have a full view of the output of the OBU and to ensure that only the minimum billing information is transmitted.

One option is to allow a device (e.g. a USB mass storage device would be sufficient) to record all data sent between the OBU unit performing the calculations, and the GSM subsystem that relays all the information back to the insurer. This solution is not invulnerable to a maliciously programmed OBU that only reveals part of the conversation. On the other hand it makes certification easier, since only a trivial property needs to hold: that all data transmitted using GSM is also recorded on the auditing device. A second approach, that offers stronger guarantees, is to physically separate (and shield) the OBU from the GSM transmitter, and link them with a recording device controlled by the user. This device would record all traffic, and allow the users to verify that the data transmitted only contains the billing information. We note that the information recorded for monitoring can be deleted straight away after the verification that no privacy leakage is happening, thus does not conflict with the privacy-strengthening solutions we discussed in the previous section.

Nevertheless, without third-party certification it is impossible to ensure that the OBU is not recording precise location data with the intent to provide them to a third party. Since such a device has no way of transmitting the recorded data over the air, physical access would be required to extract the data, making it difficult to turn this weakness into a remote surveillance tool. This is a known open problem [129], and the control of physical access would require additional certification.

Certification cannot guarantee that there are no covert channels left between the OBU and the service provider. However, even though this risk is fully eliminated, it be can reduce to a minimum. The certification goals for the OBU to provide high grades of assurance are:

- The random number generation should be based on a physical source of randomness. A pseudo-random number generator with a seed known to the provider would produce predictable encryption keys, leaving the audit logs unprotected. An alternative strategy would be for a device controlled by the user to be able to set the initial state of the random number generator.
- The deletion operation of the keys and the data in the OBU should be effective. Otherwise an adversary may be able to get access to keys and logs from the past. An alternative could be for the OBU to not hold any non-volatile memory, aside a removable memory chip – that the user can physically remove and destroy to preserve privacy.
- A thorough side channel analysis is necessary to ensure that the OBU does not leak or transmit information through any other means than the audited GSM transmission. Enclosing the OBU into a Faraday cage, using a conductive cover, could ensure this. Yet the GPS antenna, as well as the GSM module should be outside the enclosure.
- The correct implementation of the protocols procedures should be certified: the OBU only records the premium payment information; all raw location information is stored only in an encrypted form using the appropriate keys; correct slicing is performed previous to encryption; etc. This is only required to protect against adversaries with local access, since auditing minimizes the risk that personal data is transmitted remotely.

Finally, for security reasons, the provider should be able to update the software to patch bugs. For instance, the full update can be signed by a certification authority after evaluation of the new features. Such re-evaluation is expensive, and might slow down the deployment of critical security updates.

Practical issues

Although throughout the chapter we mentioned that the cost associated with roads could depend on attributes of the driver (e.g., retired users may get discounts) or on attributes of the car (e.g., ecological cars may have reduced fees), the pricing policy used by our prototype is very simple. We note that this is a limitation of the prototype and that the architecture can support more flexible policies. Also, personalized discounts can be easily integrated in the system. For instance, the TSP or the insurance company can apply discounts to the total fee reported by the OBU, without the knowledge of fine grained location data. Further, the system model in this work considers only one service provider. However, the European legislation [54, 94] points out that several TSPs may provide services in a given Toll Charger domain. PrETP can be trivially extended to this setting.

The Optimistic Payment scheme we described as part of PrETP (see Sect. 7.3.1) allows the OBU to prove its correct operation to the TSP while revealing a minimum amount of information. Nevertheless, we note that fee calculation is not flexible. The reason is that the OBU should store signatures created by the TSP on all the prices that belong to $\mathrm{Im}(f)$, and thus, for the sake of efficiency, we need to keep $\mathrm{Im}(f)$ small. For this purpose, in our evaluation f is only defined for trajectory segments of a fixed length (one kilometer) and of a fixed road type. There are two obvious cases in which this feature is problematic: when a vehicle has driven a non-integer amount of kilometers, and when one of the segments contains pieces of roads with different cost (e.g., when a driver leaves the highway entering a secondary road). Given that the policy provided by the TSP assigns a price per kilometer and type of road, no signed price for these "special" segments is available to the client? Hence, the OBU cannot produce a payment tuple.

There are two possible solutions to this problem. A first option would be to solve them at contractual level. The policy designed by the TSP could include clauses that indicate how to proceed when these conflicts arise. For instance, in the first case the TSP could dictate that the driver must pay for the whole kilometer, and in the second case the policy could be that the price corresponds to the cheapest of the roads, or to the most expensive. We note that these decisions do not conflict with the general purpose of the system: congestion control, as in all cases, on average, drivers will pay proportionally to their use of the roads. The second option would be to change the way in which the OBU proves that the committed prices belong to $\mathrm{Im}(f)$. In the construction proposed in Sect. 7.3.1, the OBU employs a set membership proof to prove that the committed prices belong to the finite set $\mathrm{Im}(f)$. Alternatively, we can define $\mathrm{Im}(f)$ as a range of (positive) prices, and let the OBU use a range proof to prove that the committed prices belong to $\mathrm{Im}(f)$. Since now $\mathrm{Im}(f)$ is much bigger, f can be defined for segments of arbitrary length that include several types of road. We outline a construction that employs range proofs in [16].

Another issue is that our OP scheme does not offer protection against OBUs that do not reply upon receiving a verification challenge. In this case, the TSP should be able to demonstrate to the TC that the OBU is misbehaving. To permit this, the TSP can delegate to the TC the verification of the "spot-check," i.e, the TSP sends the payment message m and the signature s_m to the TC, and the TC interacts with the OBU (electronically, or by contacting the driver through some other means) to verify that m is valid.

7.4.2 Methodological discussion

So far there exists little experience in how to integrate privacy enhancing technologies in the engineering of systems to be deployed in the real world. We are lacking a general methodology to define privacy requirements, as well as to find solutions to fulfill them. This is further complicated by the disconnection between systems engineers, that design and implement systems to be deployed in the real world, and the field of privacy research. New tools to build systems with strong privacy guarantees (e.g.. anonymous credentials [43], private information retrieval [50], secure multiparty computation [285], or cryptographic commitments [40]) and new findings that shake our assumptions about which are the limits of privacy protection (e.g., the impossibility of database anonymization with strong privacy guarantees [100, 202, 209]) are often ignored by systems designers.

In [133] we provide a thorough discussion on how to close the gap in privacy engineering, and on how novel research can be embedded in the design of systems. For this purpose we describe five main steps that engineers can use as guidelines in the design of privacy-preserving systems. In this section we revisit these steps and map them to the design decisions we took when building PriPAYD and PrETP. With this exercise, we aim to better illustrate the design process and improve our understanding about what is required at each of the steps.

Functional Requirements Analysis: The first step in the design of a system in which we want privacy embedded at the core is to clearly describe its functionality. That is, the goal has to be well defined and feasible. In the application which we are dealing with in this thesis, an Electronic Toll Pricing system, the functionality was clearly delimited: charge users according to when, where and how they drive (e.g., which roads they use, what time of the day they travel, etc.).

This step is a cornerstone in the path towards finding a solution that ensures strong privacy protection. Vague or implausible descriptions have a high risk of forcing engineers into a design that would collect more data, as massive data collection is needed in order to guarantee that *any* alternative realization of the system can be accommodated by the design. In the ETP case, wider functionality descriptions in which the system could be used for other purposes such as support for law enforcement or for location-based services would render our solutions useless. If data needs to be eventually available to the police (or other service providers), for purposes not clearly defined at the design stage, the local processing of location data would be ruled out of the design space. The only approach left would be to transmit all data to a centralized server that processes and distributes these data, and deals with the associated privacy risks (discussed at the beginning of this chapter).

We must stress that requiring that the functionality of a system is well-defined does not necessarily impose a limit on the system's purpose. In the ETP case the purpose of the system is simple. However, for some applications the designed solution may need to be flexible enough to integrate additional services. In these cases, these additional services also need to be articulated precisely so that their requirements can be taken into account at the early stages of the design.

Data Minimization: For a given functionality, the data that is absolutely necessary to fulfill the functionality needs to be analyzed.

In the ETP case the minimal set of data needed to tax drivers is their identity and the amount to be charged. No other private data, such as where and when the vehicle was, is strictly necessary. The service provider only needs to know the amount to charge to each of the users, regardless of their actual driving records. For instance, if Alice drives for 3km on a highway which has an assigned price of $1 per kilometer, the service provider only needs to know that Alice must pay $3 but not whether she was traveling to New York, or New Jersey.

We note that the decision as to which data is absolutely necessary for a given purpose involves a deep knowledge of the state-of-the-art research to explore which data can be minimized. In some cases, advanced privacy-preserving cryptographic techniques [40, 43, 50, 285] allow to further minimize data in a new and counter-intuitive ways. Further, advances in the cryptographic computation capabilities of the hardware platforms on which systems are built also open new possibilities for the designer. A privacy engineer must be well aware of the latest research results in order to take informed decisions that lead to the most up-to-date privacy-preserving design.

Modeling Attackers, Threats and Risks: Once the desired functionality is settled and the data that will be collected is specified, it is possible to start developing models of potential attackers, e.g., curious third parties, the service provider; the types of threats these attackers could realize, e.g., public exposure, linking, profiling.

In our design we consider that both the communication provider and the service provider are a threat for users' privacy. We then analyze the data that is available to these parties, and study the attacks on privacy that could be performed with this information.

The communication provider, as any other external adversary, cannot see the content of communications between users and the Toll Service Provider. Nevertheless, it has access to the traffic data associated to these communications and hence is in the position of performing traffic analysis to determine the location or typical trajectories of users. The service provider receives only the final fee to be paid, which in principle do not reveal sensitive information about the users. However, as we have discussed above, the adversary could still extract information if this fee is such that it could only be generated by a limited amount of trajectories.

The analysis of the likelihood and impact of the realization of the threats is not a trivial exercise. First, it requires awareness of recent research results on potential attacks and vulnerabilities. It may not always be evident which of the collected data, if any, may pose a privacy threat. Going back to the example of database de-anonymization, without the proper knowledge about the latest results on the topic [100,200–202], it may seem reasonable from a privacy point of view to collect, share and/or publish anonymized data. Finally, the analysis requires analytical expertise to extrapolate these novel results to the application under study.

Multilateral Security Requirements Analysis: Besides the system's purpose itself, the engineer must account for other constraints that ensure the security and correct behavior of the entities in the system, as expected by the different stakeholders of the system. The inclusion, analysis and resolution of these conflicting security requirements is also known as multilateral security.

In our case study the security requirements of the system imply that none of the involved parties can take advantage of the ETP system. Thus, no entity in the system (i.e., neither the service provider, nor the users) must be able to claim that a user should pay an amount different to the fee that actually corresponds to her driving records. If this requirement is fulfilled, the service provider can be sure that users do not misuse the system, while users have guarantee that they only pay for what they drive.

The goal of this analysis is to find a design in which privacy measures cannot be detrimental to other important security objectives such as integrity, availability, etc. and vice versa. For the ETP case study, the sought solution must provide means for each to check the correctness of the operations that the other entities perform, while limiting the amount of location data disclosed.

Implementation and Testing of the Design: The final step in the design of the system is to implement the solution that fulfills the multilateral security requirements revealing the minimal amount of private data. Further, the potential vulnerabilities have to be scrutinized, and the functioning of the system according to the articulated functional requirements have to be validated.

In the first step and second steps of the design approach, we concluded that the service provider does not need access to fine-grained location data. Hence, in our design we chose to place the processing of these data in the users' domain, and only communicate the final fee to the service provider. After modeling the adversaries we deal with and analyzing their capabilities, we choose to mandate that the GSM system must be only switched on when the vehicle needs to transmit data, and that this transmission must take place at a pre-determined time and location. This is done to further limit the information leakage and avoid traffic analysis on the communication data.

The most challenging aspect of the design process is the reconciliation of security and privacy requirements. In PrETP, besides protecting privacy through the local processing of sensitive information, we use cryptographic commitments in order to allow the service provider to check that these local operations have been correctly preformed. Besides the fact that minimal amount of location data are collected under normal operation, also minimal information is disclosed while answering a challenge to prove the drivers' honesty. For this purpose, location data are sliced in segments, and a sub-fee and a commitment per segment are computed. Thus, when responding to a challenge, the user only needs to disclose a small trajectory segment containing the challenged location, which is already known to the provider.

In order to complete our design, we have implemented a prototype OBU and demonstrate that, contrary to common belief, the overhead introduced by privacy enhancing technologies is moderate, and that they are efficient enough to be integrated in commercial in-vehicle devices. Finally, we have verified that the design is compliant with the legal framework in which the application is to be deployed [17, 267, 268].

We recall that there are some fundamental limits to the privacy protection offered by technical means. As a complement to our solution, in the previous section we have outlined a series of non-technical measures to mitigate privacy threats that cannot be addressed using engineering solutions.

7.5 Conclusions

Pay-as-you-drive policies present a number of advantages, are bound to gain popularity in vehicular applications. However, if care is not taken when designing these systems the resulting implementations may incur in a fundamental disregard for the privacy of vehicle owners, which might slow or even limit their deployment.

In this chapter we have proposed two systems, PriPAYD and PrETP, that support the deployment of PAYD policies while also providing strong privacy guarantees. The key principle of our designs is to avoid the need for centralized databases holding vast amounts of location data. For this purpose we push operations performed on sensitive data to a device in the user's domain. The security of this device is based on simple and well-understood multi-level security components.

Our architectures rely (as previous systems) on secure hardware for correct accounting, but privacy properties can be checked independently of the correctness of the billing just by auditing its output. This separates correct accounting from privacy concerns, allowing On-Board Units to remain fully under the control of the provider, while users can be sure that their location data is not leaking.

However, the system also needs to provide means to protect the interests of the service provider. Hence, the detection of fraud is an important requisite. In the second part of the chapter we have introduced a protocol, Optimistic Payment, that allows OBUs to prove that they operate correctly while leaking the minimum amount of information. In particular, upon request of the service provider, OBUs can attest that the location data for the calculation of the fee is authentic and has not been tampered with. While performing this attestation the OBU must reveal some location data to the service provider, but we note that at the time of disclosure these data are already known to the provider. We have defined and constructed our protocol as well as proved it secure under standard assumptions. We also provide an efficient instantiation based on known secure cryptographic primitives.

There is no component or infrastructure required by PriPAYD or PrETP that would make them significantly more expensive than their privacy-invasive alternatives, as we demonstrate with our prototype. One could in fact argue that in the long run PriPAYD and PrETP, as any other privacy enhancing technology, are cheaper than privacy-invasive systems. The costs of protecting private data stores is often overlooked in the accounting of costs, as is the risk of a single security breach leaking the location data of millions of customers [14]. In addition, our systems keep sensitive data locally in each car, in a system that is easy to engineer and verify. A back-end system that provides the same level of privacy protection to masses of data would be not only prohibitively expensive, but simply unimplementable.

At the end of the chapter we have revisited the decisions taken while building PriPAYD and PrETP, in an effort to understand the critical steps in the design of privacy-preserving systems. This dissection of the design philosophy behind our systems can be used as a reference to guide future designs. However, we must stress that the particular solutions described along this chapter are tailored to the requirements of pay-as-you-drive applications. Other applications may have different requirements, and at every step the decisions must be carefully reconsidered to find the best design and implementation.

Chapter 8

Conclusions and future work

In the last years electronic communications have become part of an increasing number of our everyday activities. Interactions between people and/or institutions are progressively being mediated by machines. This tendency continuously raises new privacy concerns forcing researchers and engineers designing systems to face new problems with diverse constraints and requirements.

The privacy community lacks a general methodology for the design and analysis of systems, and the effectiveness of privacy-preserving solutions is usually tested using ad hoc analysis specific to the system under study. This deficiency has driven the community into a disorganized arms race between designs and attacks to find the best privacy-preserving solution for each application. As a result, there is little knowledge about how to compare and validate systems in a general way, which in turn jeopardizes the development of robust privacy-enhancing designs that are ready to be integrated in real-world applications.

In this thesis we have considered the design of privacy-preserving systems from the point of view of the engineer that has to conceive a privacy-preserving solution, as well as analyze its privacy properties. In the first part of the thesis we have proposed a general methodology to quantify information leaks in anonymity systems. In the second part we proposed two privacy-preserving architectures for pay-as-you-drive services, PriPAYD and PrETP, in which security and privacy requirements are fulfilled simultaneously. Based on our experience building these applications, we have identified basic steps in the design of privacy-preserving systems. The rest of this chapter summarizes our findings, and we conclude discussing future lines of research that can extend our work.

The analysis of privacy-preserving systems

In the first part of the thesis we have presented a general methodology to model and analyze information leakage, using anonymous communications systems as a case study. Anonymous communications aim at protecting the privacy of their users by hiding who is communicating with whom. However, anonymous communication systems are known to be vulnerable to traffic analysis attacks, as we thoroughly discussed in Chapter 2. These attacks exploit various kinds of traffic information, e.g., the amount and timing of data transferred or the duration of the connection, to uncover relationships taking place over an anonymous communications network.

Our first observation, discussed in Chapter 3, is that the de-anonymization of messages is more effective when the adversary considers all users at once, rather than focusing on them individually. We present two attacks for de-anonymizing messages that outperform previous work: the Perfect Matching Disclosure Attack (PMDA) and the Normalized Statistical Disclosure Attack (NSDA). These attacks differ from each other in the underlying principle used to consider all users simultaneously. The PMDA is based on finding perfect matchings between senders and receivers of messages and, although it outputs precise results, it is computationally expensive. The NSDA relies on matrix normalization to consider interdependencies between senders and receivers. It requires less computation power than the PMDA, but it provides less accurate results. An additional advantage of our attacks is that they are robust with respect to changes in the user behavior model, as opposed to previously published Disclosure Attacks [5,60,70,76,158,161] which are optimized for a specific scenario. Further, we show that simultaneously de-anonymizing messages and estimating sender profiles yields better results than performing these tasks separately.

However, the analysis methods presented in Chapter 3 are limited. First, the straightforward manner in which we reuse information when co-inferring profiles and assignments of senders to receivers is far from optimal. Second, when considering complex systems, simultaneous estimation of profiles and sender-receiver correspondences may require a large amount of computational resources. As a solution, in Chapters 4 and 5 we propose to cast the traffic analysis problem as an inference problem, and use advanced Bayesian statistics to compute probability distributions over possible receivers of messages in an anonymity system. The techniques we present are based on sampling. Hence, they do not suffer from computational limitations, allowing us to deal with complex systems and to compute anonymity metrics in the presence of arbitrary constraints, contrary to previous results in which this was considered an intractable problem [237].

Our findings demonstrate that probabilistic modeling, Bayesian inference, and the associated conceptual toolkit relating to Markov chain Monte Carlo sampling are an appropriate basis on which to build traffic analysis attacks: i) it provides a clear framework to perform the analysis starting with the definition of a probabilistic model, that is inverted and sampled to estimate quantities of interest; ii) it ensures that information is used properly, avoiding overfitting or systematic biases; iii) it enables the analyst to answer arbitrary questions about the entities in the system with a clear probability statement; and iv) it provides good and clear estimates of error. These qualities are in sharp contrast with previous work on traffic analysis, that provides ad hoc best guesses of very specific quantities, with a separate analysis to establish their accuracy based on labeled data – something that the traffic analyst does not have when deploying attacks on the ground.

The design of privacy-preserving systems

In the second part of the thesis we presented two architectures that follow common principles to integrate strong privacy guarantees in their design. Starting from the basic functionality of the system under study (in our case pay-as-you-drive applications) we identify the minimum set of data that needs to be revealed to the service provider: the final premium to be billed. Further, we demonstrate that the fulfillment of other security requirements, as integrity or accountability, is not incompatible with the provision of strong privacy guarantees. By using advanced privacy-preserving cryptographic primitives we are able to safeguard the interests of all entities in the system while enabling users to disclose a minimum amount of personal information. In order to evaluate the suitability of our solutions for deployment in the real world we complete our design with an evaluation of its properties from a security, performance and legal perspective.

Our designs avoid the massive collection of personal information in centralized databases while providing the same functionality as data-collection-based approaches. Limiting the private information available to the provider reduces the chances of voluntary or involuntary leakage or abuse, minimizing the privacy risks inherent to the existence of these databases. Further, data collection minimization reduces the amount of information to be protected, reducing the management and maintenance cost of the database.

The design principles behind PriPAYD and PrETP are applicable to many other contexts, as for instance Smart Energy systems [228]. It is our hope that the steps taken in our design process, further elaborated in [133], lay the foundation for a general privacy engineering discipline that serves as guidance for future privacy-preserving systems designers. We must stress, however, that the specific design decisions we have taken in PriPAYD and PrETP cannot be seen as a general rule for achieving privacy protection in other applications. Our solutions limit the privacy risk by revealing the identity of the user while minimizing the amount of sensitive data disclosed to the service provider. While this choice is sufficient for the purposes of the application under study in this thesis, it is not the only approach to protect privacy. Consider the anonymous communication systems discussed in the first part of this thesis. Contrary to PriPAYD and PrETP in these systems sensitive data is disclosed and privacy is preserved by hiding the users' identity. For example, an adversary observes a cancer specialized doctor receiving messages but cannot identify the sender of those messages.

The two parts of this thesis study cases in which privacy is protected either by anonymously disclosing the transaction data, or by limiting the amount of sensitive information disclosed along with the identity. We note that these results do not represent the limits of the protection that can be offered to users. Other applications may have different requirements or constraints that allow for simultaneous anonymity and minimal disclosure of transaction data. Similarly, the development of new cryptographic primitives and protocols may allow to re-design previous solutions enhancing their privacy protection properties.

When a new solution needs to be built, it is the job of the designer to consider the requirements of the system and study the state-of-the-art in technology at the time of choosing the architecture and technologies that provide maximal privacy guarantees. This design process is specific to the application under consideration. We must stress that given the complexity of this engineering task it is not advisable to reduce privacy-enhancing design methodologies to "privacy check lists" that can easily be ticked away for compliance reasons without mitigating some of the privacy risks that a more thorough study would identify. Further, this study must not be limited to the technical part of the design but requires understanding over the legal, social, political, and economical framework in which the application has to be deployed, and the implications of these constraints on the engineered system.

Finally, even though in this thesis we have treated the analysis and design of systems as separate processes, the design and the security analysis activities have to be re-iterated to achieve maximal security. Once a system is outlined the analysis of the information leakage may uncover risks overseen at the design stage that require further minimization of data, and hence requires modifications in the proposed solution.

8.1 Future work

The analysis of privacy-preserving systems

The Bayesian treatment of long-term attacks against anonymity systems introduced in Chapter 4 is promising, but still very immature. We foresee some key theoretical, as well as implementation-related, steps to move the state of the art forward.

- The Vida Black-box model as well as the Vida Red-Blue model represent an observation from an anonymity system as a generic weighted bipartite graph, linking senders with receivers. Our experiments, on the other hand, only considered anonymity systems working in discrete rounds, in which the observation is represented by a series of full bipartite subgraphs corresponding to the rounds. This is a limitation of our sampler implementation, that could be removed in order to deal with the general case of any bipartite weighted network, as the ones studied in Chapter 5.

While in theory this modification is straightforward, in practice it is hard to directly sample matchings from arbitrary bipartite graphs. The rejection sampling algorithm suggested in Chapter 4 can be inefficient, since it might use edges that are not part of a perfect matching, forcing multiple aborts. It might be wise to first prune the assignment graph from such edges using techniques from the constraint satisfaction literature such as Regin's algorithm [225].

- Traditional hitting set as well as disclosure attacks [60,76,161] make extensive use of the number of friends of a target sender to be applicable at all, whereas the presented approaches do not require such information. Yet, adding related constraints would yield better results. The a priori model for user profiles is very general, meaning that it can represent, and thus learn, any multinomial distribution of receivers per sender. While keeping the generality, more information could be incorporated into the model, for instance if it is known to the adversary that the profiles belong to a social network (with some standard characteristics like degree, clustering etc).

- It has been an open problem in the literature how to incorporate known information about communication patterns to help the inference of unknown communication patterns. Diaz *et al.* presented in [89] an ad hoc technique to integrate social network information in the de-anonymization of traces, along with a discussion of the systematic errors that can be introduced. The sampling techniques presented in this work can be straightforwardly modified to incorporate known correspondences between senders and receivers: the Gibbs sampler is modified to only sample valid assignments that contain the known matches. These known assignments, far from being useless, drive the sampling of profiles (as part of the Gibbs sampling) leading to higher quality profiles, which in turn become higher quality assignments for the unknown messages.

The model we proposed in Chapter 5 is very rich and encompasses aspects of mix-based communications never before unified under a common framework. Its clear structure is ideal to incorporate further aspects of anonymous communications that have not been taken into account in this thesis such as:

- Other mixing strategies can be incorporated into the model besides the traditional threshold mix considered so far. The technique of Serjantov and Newman [240] can readily be adapted to model pool mixes in the current model: each round of the pool mix is represented as a separate threshold mix, where some of the messages (the pool) simply transit from one round to the next. The only modification to the current model is for this transition, from one round to another round of the same mix, not to increase the length of the path.

  More complex mix strategies require more state to be held per mix, and some of them require inference of this hidden state. Our framework is ready to accommodate such inference, effectively extending the Bayesian framework described in [207].

- Dummy messages generated by mixes to foil traffic analysis can be also incorporated in the model, by simply guessing which messages are dummies (i.e., including a flag signaling whether a message is a dummy or not in the hidden state), and describing the probability of their paths. This can be useful for foiling the protection afforded by active mixing strategies [85,207], or to model RGB-mixes [75].

- Strategies similar to the guard nodes [212] used by the Tor [93] path selection algorithm, in which only a small set of nodes per user is eligible to build the first hop of a tunnel. These strategies can be seen an as a special case of bridging, and it is trivial to incorporate them to our model.

- Finally, we have assumed that the start of all paths is known, even though the observation may be truncated before the end of the path is observed. Other models of partial network observation can also be envisaged: the adversary might just be able to observe a window of time, or only some links in the network. Models that extend the concepts of “unknown” sources or sinks of traffic can be built for these circumstances.

In this thesis we have dealt with mix networks. Nevertheless, the ‘Holy Grail’ of Bayesian traffic analysis would be its application to the setting of low-latency anonymity systems based on onion-routing [259], such as the deployed Tor [93] network, which attracts an increasing number of users. An adversary in such system is constrained to observe only a fraction of the network, but the observations leak precise cell timing data that can be used to trace streams. Murdoch and Zielinski [196] present a simplified analytical Bayesian analysis in such a setting, under the assumptions that traffic is Poisson distributed. Presenting a general model of an onion routing system, and a practical sampler to perform inference would be a significant step forward in this line of work.

The Bayesian techniques we have introduced have a strong potential to analyze privacy-preserving systems beyond anonymous communications. As long as a system has users with multinomial preferences, that are expressed and anonymized in an arbitrary manner, our algorithms are applicable to de-anonymize the preferences and extract user profiles. Thus, our approach is suitable for problems as the de-anonymization of databases [200], social networks [201] or mobile communications [26, 114, 116].

Besides its versatility to analyze systems with multiple constraints, the Bayesian framework could also be extended to consider temporal variations of the variables to infer. If the evolution of profiles over time can be modeled, as for instance in social networks [170], it can be integrated into the generative model and taken into account by the inference engine during the learning process.

The theoretical flexibility of the proposed framework makes it a strong candidate to become the standard method to evaluate anonymous communications systems. However, in practice crafting models to analyze new systems is a laborious task. Nevertheless, many of these systems have common features, e.g., routing constraints, mixing strategies, etc. Finding a way to automate the analysis of these features such that they could be combined when evaluating complex systems is a necessary step to popularize the use of Bayesian inference as the default analysis methodology.

The design of privacy-preserving systems

The design principles identified in Chapter 7 need to be backed up with advances in other fields of research. In order to obtain a proof of honest behavior from the client in the pay-as-you-drive case study we had to design a new protocol and find an efficient instantiation such that it could be implemented on a microprocessor. This is an example of an application in which even though the requirements allow for minimal disclosure of data, the technology needed for this minimization was assumed to be not available or too expensive. When new applications with new requirements emerge, new cryptographic primitives and protocols will be needed that provide wider and more flexible functionality to the designers. Further, privacy-preserving cryptographic tools are often too inefficient at the time they are proposed to be used in deployed systems. Hence, in order to easily integrate privacy-enhancing technologies in real-world systems the advances in cryptography must be followed by research that provides fast, small, and inexpensive implementations appropriate for their use in commercial products.

The lessons learned while designing PriPAYD and PrETP are valuable, but serve only as example of how to embed privacy in the design of systems. In order to develop a general methodology for the design of privacy-preserving systems more applications have to be studied to identify the critical activities in the design process. We take a first step in this direction in [133], where we study a second use case: a privacy-preserving e-petition system, in which user's privacy is guaranteed by hiding their identity from the provider while revealing their votes [83]. Our study shows that, even though these applications differ considerably in their requirements and the nature of their privacy-preserving solutions, there are many common activities in the design of these solutions. More case studies are needed to refine the description of the activities described in [133].

Finally, it has been pointed out by Gürses that eliciting privacy requirements is not an easy task due to the subjective nature of privacy itself [134]. The knowledge gathered by studying new use cases can also be useful to extend and refine the methodology for eliciting privacy requirements in [134], which in turn shall ease the task of engineering privacy-preserving systems.

Bibliography

[1] Twitter. http://twitter.com/. Accessed March 2011.

[2] Wikileaks. http://www.wikileaks.org/. Accessed March 2011.

[3] aka-aki. http://www.aka-aki.com/, 2010. Accessed March 2011.

[4] Loopt. http://www.loopt.com/, 2010. Accessed March 2011.

[5] Dakshi Agrawal and Dogan Kesdogan. Measuring anonymity: The disclosure attack. *IEEE Security & privacy*, 1(6):27–34, 2003.

[6] Aioi. http://www.aioinissaydowa.co.jp/english/, 2010. Accessed March 2011.

[7] Ross Anderson, Mike Bond, Jolyon Clulow, and Sergei Skorobogatov. Cryptographic Processors - A survey. *Proceedings of the IEEE*, 94(2):357–369, 2006.

[8] Ross J. Anderson. *Security engineering.* Wiley New York, 2nd edition, 2001.

[9] Ross J. Anderson, Serge Vaudenay, Bart Preneel, and Kaisa Nyberg. The Newton channel. In Ross J. Anderson, editor, *1st International Workshop on Information Hiding (IH 1996)*, volume 1174 of *Lecture Notes in Computer Science*, pages 151–156. Springer, 1996.

[10] Claudio Agostino Ardagna, Marco Cremonini, Ernesto Damiani, Sabrina De Capitani di Vimercati, and Pierangela Samarati. Location privacy protection through obfuscation-based techniques. In Steve Barker and Gail-Joon Ahn, editors, *21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security (DBSec 2007)*, volume 4602 of *Lecture Notes in Computer Science*, pages 47–60. Springer, 2007.

[11] ARM. ARM7TDMI technical reference manual, revision: r4p3. http://infocenter.arm.com/help/topic/com.arm.doc.ddi0234b/DDI0234.pdf, 2004. Accessed March 2011.

[12] American Automobile Association. `http://www.aaa.com/`. Accessed March 2011.

[13] National Motorist Association. NMA's position on auto insurance. `http://www.motorists.org`, 1998. Accessed March 2011.

[14] UK Financial Services Authority. FSA fines Norwich Union Life £1.26m for exposing its customers to the risk of fraud. `http://www.fsa.gov.uk/pages/Library/Communication/PR/2007/130.shtml`, December 2007. Accessed March 2011.

[15] Adam Back, Ulf Möller, and Anton Stiglic. Traffic analysis attacks and trade-offs in anonymity providing systems. In Ira S. Moskowitz, editor, *4th International Workshop on Information Hiding (IH 2001)*, volume 2137 of *Lecture Notes in Computer Science*, pages 245–257. Springer, 2001.

[16] Josep Balasch, Alfredo Rial, Carmela Troncoso, Christophe Geuens, Bart Preneel, and Ingrid Verbauwhede. PrETP: Privacy-preserving electronic toll pricing (extended version). Technical report, Katholieke Universiteit Leuven, 2010.

[17] Josep Balasch, Alfredo Rial, Carmela Troncoso, Bart Preneel, Ingrid Verbauwhede, and Christophe Geuens. PrETP: Privacy-preserving electronic toll pricing. In *19th USENIX Security Symposium*, pages 63–78. USENIX Association, 2010.

[18] Josep Balasch, Ingrid Verbauwhede, and Bart Preneel. An embedded platform for privacy-friendly road charging applications. In *Design, Automation and Test in Europe (DATE 2010)*, pages 867–872. IEEE, 2010.

[19] Bhuvan Bamba, Ling Liu, Péter Pesti, and Ting Wang. Supporting anonymous location queries in mobile environments with PrivacyGrid. In Jinpeng Huai, Robin Chen, Hsiao-Wuen Hon, Yunhao Liu, Wei-Ying Ma, Andrew Tomkins, and Xiaodong Zhang, editors, *17th International Conference on World Wide Web (WWW 2008)*, pages 237–246. ACM, 2008.

[20] Jaroslav Ban. Cryptographic library for ARM7TDMI processors. Master's thesis, T.U. Kosice, 2007.

[21] Albert-László Barabási and Eric Bonabeau. Scale-free networks. *Scientific American*, 288(5):50–59, 2003.

[22] Kevin S. Bauer, Damon McCoy, Dirk Grunwald, Tadayoshi Kohno, and Douglas C. Sicker. Low-resource routing attacks against Tor. In Peng Ning and Ting Yu, editors, *ACM Workshop on Privacy in the Electronic Society (WPES 2007)*, pages 11–20. ACM, 2007.

[23] David Elliott Bell and Leonard J. LaPadula. *Secure Computer Systems: Mathematical Foundations and Model.* Mitre, 1974.

[24] Paolo Bellavista, Antonio Corradi, and Carlo Giannelli. Efficiently managing location information with privacy requirements in Wi-Fi networks: a middleware approach. In *2nd International Symposium of Wireless Communication Systems 2005 (ISWCS 2005)*, pages 1–8. IEEE, 2005.

[25] Richard Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16:87–90, 1958.

[26] Alastair R. Beresford and Frank Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.

[27] Daniel J. Bernstein. The salsa20 family of stream ciphers. In Matthew J. B. Robshaw and Olivier Billet, editors, *New Stream Cipher Designs - The eSTREAM Finalists*, volume 4986 of *Lecture Notes in Computer Science*, pages 84–97. Springer, 2008.

[28] Oliver Berthold, Hannes Federrath, and Stefan Köpsell. Web mixes: A system for anonymous and unobservable internet access. In Hannes Federrath, editor, *Designing Privacy Enhancing Technologies, International Workshop on Design Issues in Anonymity and Unobservability*, volume 2009 of *Lecture Notes in Computer Science*, pages 115–129. Springer, 2000.

[29] Oliver Berthold and Heinrich Langos. Dummy traffic against long term intersection attacks. In Roger Dingledine and Paul F. Syverson, editors, *2nd International Workshop on Privacy Enhancing Technologies (PET 2002)*, volume 2482 of *Lecture Notes in Computer Science*, pages 110–128. Springer, 2002.

[30] Oliver Berthold, Andreas Pfitzmann, and Ronny Standtke. The disadvantages of free MIX routes and how to overcome them. In Hannes Federrath, editor, *Designing Privacy Enhancing Technologies, International Workshop on Design Issues in Anonymity and Unobservability,*, volume 2009 of *Lecture Notes in Computer Science*, pages 30–45. Springer, 2000.

[31] Claudio Bettini, Xiaoyang Sean Wang, and Sushil Jajodia. Protecting privacy against location-based personal identification. In Willem Jonker and Milan Petkovic, editors, *Second VLDB Workshop on Secure Data Management (SDM 2005)*, volume 3674 of *Lecture Notes in Computer Science*, pages 185–199. Springer, 2005.

[32] Alex Biryukov and Eyal Kushilevitz. From differential cryptoanalysis to ciphertext-only attacks. In Hugo Krawczyk, editor, *Advances in Cryptology (CRYPTO '98)*, volume 1462 of *Lecture Notes in Computer Science*, pages 72–88. Springer, 1998.

[33] Subir Biswas, Raymond Tatchikou, and Francois Dion. Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety. *IEEE Communication Magazine*, 44(1):74–82, 2006.

[34] Jean-François Blanchette and Deborah G. Johnson. Data Retention and the Panoptic Society: The Social Benefits of Forgetfulness. *The Information Society*, 18(1):33–45, 2002.

[35] Andrew Blumberg and Robin Chase. Congestion privacy that respects "Driver Privacy". In *IEEE International Conference on Intelligent Transportation Systems (ITSC 2005)*. IEEE, 2005.

[36] Andrew Blumberg, Lauren Keeler, and Abhi Shelat. Automated traffic enforcement which respects driver privacy. In *IEEE International Conference on Intelligent Transportation Systems (ITSC 2004)*, 2004.

[37] Andrew J. Blumberg and Peter Eckersley. On locational privacy, and how to avoid losing it forever. Technical report, Electronic Frontier Foundation, 2009.

[38] Nikita Borisov, George Danezis, Prateek Mittal, and Parisa Tabriz. Denial of service or denial of security? In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *ACM Conference on Computer and Communications Security (CCS 2007)*, pages 92–102. ACM, 2007.

[39] Philippe Boucher, Adam Shostack, and Ian Goldberg. Freedom systems 2.0 architecture. White paper, Zero Knowledge Systems, Inc., 2000.

[40] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

[41] Levente Buttyán, Tamás Holczer, and István Vajda. On the effectiveness of changing pseudonyms to provide location privacy in VANETs. In Frank Stajano, Catherine Meadows, Srdjan Capkun, and Tyler Moore, editors, *4th European Workshop Security and Privacy in Ad-hoc and Sensor Networks (ESAS 2007)*, volume 4572 of *Lecture Notes in Computer Science*, pages 129–141. Springer, 2007.

[42] Giorgio Calandriello, Panos Papadimitratos, Jean-Pierre Hubaux, and Antonio Lioy. Efficient and robust pseudonymous authentication in VANET. In Wieland Holfelder, Paolo Santi, Yih-Chun Hu, and Jean-Pierre Hubaux, editors, *4th International Workshop on Vehicular Ad Hoc Networks (VANET 2007)*, pages 19–28. ACM, 2007.

[43] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In

Birgit Pfitzmann, editor, *Advances in Cryptology - EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118. Springer, 2001.

[44] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano, editors, *Security in Communication Networks (SCN 2002)*, volume 2576 of *Lecture Notes in Computer Science*, pages 268–289. Springer, 2002.

[45] Kathleen M. Carley. Destabilization of covert networks. *Computational & Mathematical Organization Theory*, 12(1):51–66, 2006.

[46] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Prakash Panangaden. Probability of error in information-hiding protocols. In *20th IEEE Computer Security Foundations Symposium (CSF 2007)*, pages 341–354. IEEE Computer Society, 2007.

[47] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[48] Siddhartha Chib and Edward Greenberg. Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 49(4):327–335, 1995.

[49] Eun-Ae Cho, Chang-Joo Moon, Hyun-Soo Im, and Doo-Kwon Baik. An anonymous communication model for privacy-enhanced location based service using an echo agent. In Won Kim, Hyung-Jin Choi, and Dongho Won, editors, *3rd International Conference on Ubiquitous Information Management and Communication (ICUIMC 2009)*, pages 290–297. ACM, 2009.

[50] Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *Journal of the ACM*, 45(6):965–981, 1998.

[51] Richard Chow and Philippe Golle. Faking contextual data for fun, profit, and privacy. In Stefano Paraboschi and Ehab Al-Shaer, editors, *Proceedings of the 8th ACM workshop on Privacy in the electronic society (WPES 2009)*, pages 105–108. ACM, 2009.

[52] Sebastian Clauß and Stefan Schiffner. Structuring anonymity metrics. In Ari Juels, Marianne Winslett, and Atsuhiro Goto, editors, *Workshop on Digital Identity Management (DIM 2006)*, pages 55–62. ACM, 2006.

[53] Metropolitan Transportation Comission. AB 744 (Torrico) – Authorize a Bay Area Express Lane Network to Deliver Congestion Relief and Public Transit Funding with No New Taxes, August 2009.

[54] Commission Decission of 6 October 2009 on the definition of the European Electronic Toll Service and its technical elements, 2009.

[55] Cory Cornelius, Apu Kapadia, David Kotz, Daniel Peebles, Minho Shin, and Nikos Triandopoulos. Anonysense: privacy-aware people-centric sensing. In Dirk Grunwald, Richard Han, Eyal de Lara, and Carla Schlatter Ellis, editors, *6th International Conference on Mobile Systems, Applications, and Services (MobiSys 2008)*, pages 211–224. ACM, 2008.

[56] Enrique Costa-Montenegro, Carmela Troncoso, Claudia Diaz, and Stefan Schiffner. On the difficulty of achieving anonymity for Vehicle-2-X communication. *Accepted at Comput. Netw. Special issue on Deploying Vehicle-2-X Communication*, page 25, 2011.

[57] Coverbox Wunelli Limited. `http://www.coverbox.co.uk/`. Accessed March 2011.

[58] Michael R. Curry and David Phillips. Privacy and the phenetic urge: geodemographics and the changing spatiality of local practice. In David Lyon, editor, *Surveillance as Social Sorting: Privacy, Risk, and Automated Discrimination.* London: Routledge, 2003.

[59] Ivan Damgård and Eiichiro Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In Yuliang Zheng, editor, *Advances in Cryptology (ASIACRYPT 2002)*, volume 2501 of *Lecture Notes in Computer Science*, pages 125–142. Springer, 2002.

[60] George Danezis. Statistical disclosure attacks: Traffic confirmation in open environments. In Gritzalis, Vimercati, Samarati, and Katsikas, editors, *Proceedings of Security and Privacy in the Age of Uncertainty, (SEC2003)*, pages 421–426, Athens, May 2003. IFIP TC11, Kluwer.

[61] George Danezis. The traffic analysis of continuous-time mixes. In *Proceedings of Privacy Enhancing Technologies workshop (PET 2004)*, volume 3424 of *Lecture Notes in Computer Science*, pages 35–50, May 2004.

[62] George Danezis. Breaking four mix-related schemes based on universal re-encryption. In Sokratis K. Katsikas, Javier Lopez, Michael Backes, Stefanos Gritzalis, and Bart Preneel, editors, *9th International Conference on Information Security (ISC 2006)*, volume 4176 of *Lecture Notes in Computer Science*, pages 46–59. Springer, 2006.

[63] George Danezis. Breaking four mix-related schemes based on universal re-encryption. *International Journal of Information Security*, 6(6):393–402, 2007.

[64] George Danezis and Richard Clayton. Route fingerprinting in anonymous communications. In Alberto Montresor, Adam Wierzbicki, and Nahid Shahmehri, editors, *International Conference on Peer-to-Peer Computing (P2P 2006)*, pages 69–72. IEEE Computer Society, 2006.

[65] George Danezis and Richard Clayton. Introducing traffic analysis. In Alessandro Acquisti, Stefanos Gritzalis adn Costas Lambrinoudakis, and Sabrina di Vimercati, editors, *Digital Privacy: Theory, Technologies and Practices*, pages 95–117. Auerbach Publications, 2007.

[66] George Danezis and Claudia Diaz. Space-efficient private search with applications to rateless codes. In Sven Dietrich and Rachna Dhamija, editors, *11th International Conference on Financial Cryptography and Data Security (FC 2007)*, volume 4886 of *Lecture Notes in Computer Science*, pages 148–162. Springer, 2007.

[67] George Danezis, Claudia Diaz, Sebastian Faust, Emilia Käsper, Carmela Troncoso, and Bart Preneel. Efficient negative databases from cryptographic hash functions. In Juan A. Garay, Arjen K. Lenstra, Masahiro Mambo, and René Peralta, editors, *10th International Conference on Information Security (ISC 2007)*, volume 4779 of *Lecture Notes in Computer Science*, pages 423–436. Springer, 2007.

[68] George Danezis, Claudia Diaz, Emilia Käsper, and Carmela Troncoso. The wisdom of crowds: Attacks and optimal constructions. In Michael Backes and Peng Ning, editors, *14th European Symposium on Research in Computer Security (ESORICS 2009)*, volume 5789 of *Lecture Notes in Computer Science*, pages 406–423. Springer, 2009.

[69] George Danezis, Claudia Diaz, and Paul Syverson. Systems for anonymous communication. In Burton Rosenberg, editor, *Handbook of Financial Cryptography and Security*, Cryptography and Network Security Series, pages 341–389. Chapman & Hall/CRC, 2009.

[70] George Danezis, Claudia Diaz, and Carmela Troncoso. Two-sided statistical disclosure attack. In Nikita Borisov and Philippe Golle, editors, *7th International Symposium on Privacy Enhancing Technologies (PETS 2007)*, volume 4776 of *Lecture Notes in Computer Science*, pages 30–44. Springer-Verlag, 2007.

[71] George Danezis, Claudia Diaz, Carmela Troncoso, and Ben Laurie. Drac: An architecture for anonymous low-volume communications. In Mikhail J. Atallah and Nicholas J. Hopper, editors, *10th International Symposium on Privacy Enhancing Technologies (PETS 2010)*, volume 6205 of *Lecture Notes in Computer Science*, pages 202–219. Springer, 2010.

[72] George Danezis, Roger Dingledine, and Nick Mathewson. Mixminion: Design of a Type III Anonymous Remailer Protocol. In *IEEE Symposium on Security and Privacy (S&P 2003)*, pages 2–15. IEEE Computer Society, 2003.

[73] George Danezis and Ben Laurie. Minx: a simple and efficient anonymous packet format. In Vijay Atluri, Paul F. Syverson, and Sabrina De Capitani di Vimercati, editors, *ACM Workshop on Privacy in the Electronic Society (WPES 2004)*, pages 59–65. ACM, 2004.

[74] George Danezis and Prateek Mittal. SybilInfer: Detecting Sybil nodes using social networks. In *Network and Distributed System Security Symposium, (NDSS 2009)*. The Internet Society, 2009.

[75] George Danezis and Len Sassaman. Heartbeat traffic to counter (n-1) attacks. In Sushil Jajodia, Pierangela Samarati, and Paul F. Syverson, editors, *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2003)*, pages 89–93. ACM, 2003.

[76] George Danezis and Andrei Serjantov. Statistical disclosure or intersection attacks on anonymity systems. In Jessica J. Fridrich, editor, *6th International Workshop on Information Hiding (IH 2004)*, volume 3200 of *Lecture Notes in Computer Science*, pages 293–308. Springer, 2004.

[77] George Danezis and Paul F. Syverson. Bridging and fingerprinting: Epistemic attacks on route selection. In Nikita Borisov and Ian Goldberg, editors, *8th Privacy Enhancing Technologies Symposium (PETS 2008)*, volume 5134 of *Lecture Notes in Computer Science*, pages 151–166. Springer, 2008.

[78] George Danezis and Carmela Troncoso. The application of Bayesian inference to traffic analysis. Technical report, Micorsoft Research, December 2008.

[79] George Danezis and Carmela Troncoso. Vida: How to use Bayesian inference to de-anonymize persistent communications. In Ian Goldberg and Mikhail J. Atallah, editors, *9th Privacy Enhancing Technologies Symposium (PETS 2009)*, volume 5672 of *Lecture Notes in Computer Science*, pages 56–72. Springer, 2009.

[80] Wiebren de Jonge and Bart Jacobs. Privacy-friendly electronic traffic pricing via commits. In Pierpaolo Degano, Joshua Guttman, and Fabio Martinelli, editors, *5th International Workshop on Formal Aspects in Security and Trust (FAST 2008)*, volume 5491 of *Lecture Notes in Computer Science*, pages 143–161. Springer, 2008.

[81] Yuxin Deng, Jun Pang, and Peng Wu. Measuring anonymity with relative entropy. In Theodosis Dimitrakos, Fabio Martinelli, Peter Y. A. Ryan, and Steve A. Schneider, editors, *4th International Workshop on Formal Aspects in Security and Trust (FAST 2006)*, volume 4691 of *Lecture Notes in Computer Science*, pages 65–79. Springer, 2006.

[82] Claudia Diaz, Joris Claessens, Stefaan Seys, and Bart Preneel. Information theory and anonymity. In B. Macq and J.-J. Quisquater, editors, *Werkgemeenschap voor Informatie en Communicatietheorie*, pages 179–186, 2002.

[83] Claudia Diaz, Eleni Kosta, Hannelore Dekeyser, Markulf Kohlweiss, and Girma Nigusse. Privacy preserving electronic petitions. *Identity in the Information Society*, 1(1):203–209, 2009.

[84] Claudia Diaz, Steven J. Murdoch, and Carmela Troncoso. Impact of network topology on anonymity and overhead in low-latency anonymity networks. In Mikhail J. Atallah and Nicholas J. Hopper, editors, *10th International Symposium (PETS 2010)*, volume 6205 of *Lecture Notes in Computer Science*, pages 184–201. Springer, 2010.

[85] Claudia Diaz and Bart Preneel. Reasoning about the anonymity provided by pool mixes that generate dummy traffic. In Jessica J. Fridrich, editor, *6th International Workshop on Information Hiding (IH 2004)*, volume 3200 of *Lecture Notes in Computer Science*, pages 309–325. Springer, 2004.

[86] Claudia Diaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. In Roger Dingledine and Paul F. Syverson, editors, *2nd International Workshop on Privacy Enhancing Technologies (PET 2002)*, volume 2482 of *Lecture Notes in Computer Science*, pages 54–68. Springer, 2002.

[87] Claudia Diaz, Carmela Troncoso, and George Danezis. Does additional information always reduce anonymity? In Ting Yu, editor, *Workshop on Privacy in the Electronic Society 2007*, pages 72–75. ACM, 2007.

[88] Claudia Diaz, Carmela Troncoso, and Bart Preneel. A framework for the analysis of mix-based steganographic file systems. In Sushil Jajodia and Javier López, editors, *13th European Symposium on Research in Computer Security (ESORICS 2008)*, volume 5283 of *Lecture Notes in Computer Science*, pages 428–445. Springer, 2008.

[89] Claudia Diaz, Carmela Troncoso, and Andrei Serjantov. On the impact of social network profiling on anonymity. In Nikita Borisov and Ian Goldberg, editors, *Proceedings of the Eighth International Symposium on Privacy Enhancing Technologies (PETS 2008)*, pages 44–62. Springer-Verlag, July 2008.

[90] Whitfield Diffie and Susan Landau. *Privacy on the line - the politics of wiretapping and encryption (updated expanded ed.)*. MIT Press, 2007.

[91] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[92] Vassil S. Dimitrov, Graham A. Jullien, and William C. Miller. Complexity and fast algorithms for multiexponentiations. *IEEE Transactions on Computers*, 49(2), 2000.

[93] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, pages 303–320. USENIX, 2004.

[94] Directive 2004/52/EC of the European Parliament and of the Council of 29 April 2004 on the interoperability of electronic road toll systems in the Community, 2004.

[95] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data, 1995.

[96] Hans Dobbertin, Antoon Bosselaers, and Bart Preneel. RIPEMD-160: A Strengthened Version of RIPEMD. In Dieter Gollmann, editor, *Fast Software Encryption (FSE 96)*, volume 1039 of *Lecture Notes in Computer Science*, pages 71–82. Springer, 1996.

[97] Matthew J. Dombroski and Kathleen M. Carley. NETEST: Estimating a terrorist network's structure. *Computational & Mathematical Organization Theory*, 8(3):235–241, 2002.

[98] Dopplr. `http://www.dopplr.com/`. Accessed March 2011.

[99] Matt Duckham and Lars Kulik. A formal model of obfuscation and negotiation for location privacy. In Hans-Werner Gellersen, Roy Want, and Albrecht Schmidt, editors, *3rd International Conference Pervasive Computing (Pervasive 2005)*, volume 3468 of *Lecture Notes in Computer Science*, pages 152–170. Springer, 2005.

[100] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *33rd International Colloquium on Automata, Languages and Programming (ICALP 2006)*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006.

[101] Morris Dworkin. Recommendation for block cipher modes of operation: The CCM mode for authentication and confidentiality. NIST special publication 800-38c, National Institute for Standards and Technology, 2004.

[102] Nathan Eagle and Alex Pentland. `http://reality.media.mit.edu/serendipity.php`, 2010. Accessed March 2011.

[103] Matthew Edman, Fikret Sivrikaya, and Bülent Yener. A combinatorial approach to measuring anonymity. In *IEEE International Conference on Intelligence and Security Informatics (ISI 2007)*, pages 356–363. IEEE, 2007.

[104] Matthew Edman and Bülent Yener. On anonymity in an electronic society: A survey of anonymous communication systems. *ACM Computing Surveys*, 42(1), 2010.

[105] Shane B. Eisenman, Emiliano Miluzzo, Nicholas D. Lane, Ronald A. Peterson, Gahng-Seop Ahn, and Andrew T. Campbell. BikeNet: A mobile sensing system for cyclist experience mapping. *ACM Transactions on Sensor Networks*, 6(1):15, 2009.

[106] Computer Engineering and Networks Laboratory ETH Zurich. Blue*. `http://www.csg.ethz.ch/research/projects/Blue_star`, 2010. Accessed March 2011.

[107] Paul Erdös and Albert Rényi. On the evolution of random graphs. *Publications of the Mathematical Institute Hungarian Academy of Sciences*, 5:17–61, 1960.

[108] Alberto Escudero-Pascual and Ian Hosein. Questioning lawful access to traffic data. *Communications of the ACM*, 47(3):77–82, 2004.

[109] Fernando Esponda, Elena S. Ackley, Paul Helman, Haixia Jia, and Stephanie Forrest. Protecting data privacy through hard-to-reverse negative databases. In Sokratis K. Katsikas, Javier Lopez, Michael Backes, Stefanos Gritzalis, and Bart Preneel, editors, *9th International Conference on Information Security (ISC 2006)*, volume 4176 of *Lecture Notes in Computer Science*, pages 72–84. Springer, 2006.

[110] Insurance Institute for Highway Safety. State automated enforcement laws, 2005.

[111] Free Software Foundation. GMP: The GNU multi-precision library.

[112] Michael J. Freedman and Robert Morris. Tarzan: a peer-to-peer anonymizing network layer. In Vijayalakshmi Atluri, editor, *9th ACM Conference on Computer and Communications Security (CCS 02)*, pages 193–206. ACM, 2002.

[113] Julien Freudiger, Mohammad Hossein Manshaei, Jean-Pierre Hubaux, and David C. Parkes. On non-cooperative location privacy: a game-theoretic analysis. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *ACM Conference on Computer and Communications Security (CCS 2009)*, pages 324–337. ACM, 2009.

[114] Julien Freudiger, Maxim Raya, Márk Félegyházi, Panos Papadimitratos, and Jean-Pierre Hubaux. Mix-Zones for Location Privacy in Vehicular Networks. In *ACM Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS 2007)*, Vancouver, 2007. ACM.

[115] Julien Freudiger, Maxim Raya, and Jean-Pierre Hubaux. Self-Organized Anonymous Authentication in Mobile Ad Hoc Networks. In *Conference on Security and Privacy in Communication Networks (Securecomm 2009)*, pages 350–372, 2009.

[116] Julien Freudiger, Reza Shokri, and Jean-Pierre Hubaux. On the optimal placement of mix zones. In Ian Goldberg and Mikhail J. Atallah, editors, *Privacy Enhancing Technologies Symposium (PETS 2009)*, volume 5672 of *Lecture Notes in Computer Science*, pages 216–234. Springer, 2009.

[117] Bugra Gedik and Ling Liu. Location privacy in mobile systems: A personalized anonymization model. In *25th International Conference on Distributed Computing Systems (ICDCS 2005)*, pages 620–629. IEEE Computer Society, 2005.

[118] Bugra Gedik and Ling Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing*, 7(1):1–18, 2008.

[119] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis, Second Edition.* Chapman & Hall/CRC, 2003.

[120] Gabriel Ghinita, Panos Kalnis, Ali Khoshgozaran, Cyrus Shahabi, and Kian-Lee Tan. Private queries in location based services: anonymizers are not necessary. In Jason Tsong-Li Wang, editor, *ACM SIGMOD International Conference on Management of Data (SIGMOD 2008)*, pages 121–132. ACM, 2008.

[121] Benedikt Gierlichs, Carmela Troncoso, Claudia Diaz, Bart Preneel, and Ingrid Verbauwhede. Revisiting a combinatorial approach toward measuring anonymity. In Marianne Winslett, editor, *Workshop on Privacy in the Electronic Society (WPES 2008)*, pages 111–116. ACM, 2008.

[122] Virgil D. Gligor. *A Guide to Understanding Covert Channel Analysis of Trusted Systems.* National Computer Security Center, ncsc-tg-030 version-1 edition, 1993.

[123] David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. Hiding routing information. In Ross J. Anderson, editor, *1st International Workshop on Information Hiding*, volume 1174 of *Lecture Notes in Computer Science*, pages 137–150. Springer, 1996.

[124] David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. Onion routing for anonymous and private internet connections. *Communications of the ACM*, 42(2):39–41, 1999.

[125] James K. Goldston. *A guide to understanding data remanence in automated information systems [microform].* NCSC-TG ; 025. National Computer Security Center, [Fort George G. Meade, MD], version 2. edition, 1991.

[126] Shafi Goldwasser, Silvio Micali, and Ron Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.

[127] Philippe Golle and Kurt Partridge. On the anonymity of home/work location pairs. In Hideyuki Tokuda, Michael Beigl, Adrian Friday, A. J. Bernheim Brush, and Yoshito Tobe, editors, *Pervasive*, volume 5538 of *Lecture Notes in Computer Science*, pages 390–397. Springer, 2009.

[128] Google. `https://www.google.com/latitude/`, 2010. Accessed March 2011.

[129] Vanessa Gratzer and David Naccache. Alien *s.* quine, the vanishing circuit and other tales from the industry's crypt. In Serge Vaudenay, editor, *25th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2006)*, volume 4004 of *Lecture Notes in Computer Science*, pages 48–58. Springer, 2006.

[130] Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *1st International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 31–42. USENIX, 2003.

[131] Marco Gruteser and Baik Hoh. On the anonymity of periodic location samples. In Dieter Hutter and Markus Ullmann, editors, *Security in Pervasive Computing, Second International Conference (SPC)*, volume 3450 of *Lecture Notes in Computer Science*, pages 179–192. Springer, 2005.

[132] Marco Gruteser and Xuan Liu. Protecting privacy in continuous location-tracking applications. *IEEE Security & Privacy*, 2(2):28–34, 2004.

[133] Seda Güerses, Carmela Troncoso, and Claudia Diaz. Engineering Privacy by Design (extended abstract). In *4th International Conference on Computers, Privacy & Data Protection (CPDP 2011)*, page 25. Springer, 2011.

[134] Seda F. Gürses. *Multilateral Privacy Requirements Analysis in Online Social Networks.* PhD thesis, Katholieke Universiteit Leuven, 2010.

[135] Serge Gutwirth. *Privacy and the Information Age.* Rowman & Littlefield Publishers, 2002.

[136] W. Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, April 1970.

[137] Urs Hengartner. Hiding location information from location-based services. In Christian Becker, Christian S. Jensen, Jianwen Su, and Daniela Nicklas, editors, *8th International Conference on Mobile Data Management (MDM 2007)*, pages 268–272. IEEE, 2007.

[138] Michael Herman. *Intelligence power in peace and war*. Cambridge University Press, 1996.

[139] Dominik Herrmann, Rolf Wendolsky, and Hannes Federrath. Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naive-bayes classifier. In *Proceedings of the 2009 ACM workshop on Cloud computing security (CCSW '09)*, pages 31–42. ACM, 2009.

[140] Theodore P. Hill. The difficulty of faking data. *Chance*, 12:27–31, January 1999.

[141] Andrew Hintz. Fingerprinting websites using traffic analysis. In Roger Dingledine and Paul F. Syverson, editors, *2nd International Workshop Privacy Enhancing Technologies (PET 02)*, volume 2482 of *Lecture Notes in Computer Science*, pages 171–178. Springer, 2002.

[142] Baik Hoh, Marco Gruteser, Ryan Herring, Jeff Ban, Daniel B. Work, Juan Carlos Herrera, Alexandre M. Bayen, Murali Annavaram, and Quinn Jacobson. Virtual trip lines for distributed privacy-preserving traffic monitoring. In Dirk Grunwald, Richard Han, Eyal de Lara, and Carla Schlatter Ellis, editors, *6th International Conference on Mobile Systems, Applications, and Services (MobiSys 2008)*, pages 15–28. ACM, 2008.

[143] Baik Hoh, Marco Gruteser, Hui Xiong, and Ansaf Alrabady. Preserving privacy in GPS traces via uncertainty-aware path cloaking. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *ACM Conference on Computer and Communications Security (CCS 2007)*, pages 161–171. ACM, 2007.

[144] Baik Hoh, Marco Gruteser, Hui Xiong, and Ansaf Alrabady. Achieving guaranteed anonymity in GPS traces via uncertainty-aware path cloaking. *IEEE Transactions on Mobile Computing*, 9(8):1089–1107, 2010.

[145] Nicholas Hopper, Eugene Y. Vasserman, and Eric Chan-Tin. How much anonymity does network latency leak? In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *ACM Conference on Computer and Communications Security (CCS 2007)*, pages 82–91. ACM, 2007.

[146] Leping Huang, Hiroshi Yamane, Kanta Matsuura, and Kaoru Sezaki. Towards modeling wireless location privacy. In George Danezis and David Martin, editors, *5th Privacy Enhancing Technologies Workshop (PET 2005)*,

volume 3856 of *Lecture Notes in Computer Science*, pages 59–77. Springer, 2005.

[147] Leping Huang, Hiroshi Yamane, Kanta Matsuura, and Kaoru Sezaki. Silent cascade: Enhancing location privacy without communication qos degradation. In John A. Clark, Richard F. Paige, Fiona Polack, and Phillip J. Brooke, editors, *Security in Pervasive Computing (SPC 2006)*, volume 3934 of *Lecture Notes in Computer Science*, pages 165–180. Springer, 2006.

[148] Dominic Hughes and Vitaly Shmatikov. Information hiding, anonymity and privacy: A modular approach. *Journal of Computer Security*, 12(1):3–36, 2004.

[149] Bret Hull, Vladimir Bychkovsky, Yang Zhang, Kevin Chen, Michel Goraczko, Allen Miu, Eugene Shih, Hari Balakrishnan, and Samuel Madden. Cartel: a distributed mobile sensor computing system. In Andrew T. Campbell, Philippe Bonnet, and John S. Heidemann, editors, *4th International Conference on Embedded Networked Sensor Systems (SenSys 2006)*, pages 125–138. ACM, 2006.

[150] Hollard Insurance. Pay as you Drive! `http://www.payasyoudrive.co.za/`. Accessed March 2011.

[151] Muhammad Iqbal and Samsung Lim. An automated real-world privacy assessment of GPS tracking and profiling. In *2nd Workshop on Social Implications of National Security: From Dataveillance to Uberveillance*, pages 225–240, 2007.

[152] Markus Jakobsson, Ari Juels, and Ronald L. Rivest. Making mix nets robust for electronic voting by randomized partial checking. In Dan Boneh, editor, *11th USENIX Security Symposium,*, pages 339–353. USENIX, 2002.

[153] Edwin Thompson Jaynes. *Probability Theory: The Logic of Science.* Cambridge University Press, 2003.

[154] Carter Jernigan and Behram F. T. Mistree. Gaydar: Facebook friendships expose sexual orientation. *First Monday*, 14(10), 2009.

[155] Panos Kalnis, Gabriel Ghinita, Kyriakos Mouratidis, and Dimitris Papadias. Preventing location-based identity inference in anonymous spatial queries. *IEEE Transactions on Knowledge and Data Engineering*, 19(12):1719–1733, 2007.

[156] Apu Kapadia, Nikos Triandopoulos, Cory Cornelius, Daniel Peebles, and David Kotz. AnonySense: Opportunistic and privacy-preserving context collection. In Jadwiga Indulska, Donald J. Patterson, Tom Rodden, and Max Ott, editors, *6th International Conference on Pervasive Computing*

(Pervasive 2008), volume 5013 of *Lecture Notes in Computer Science*, pages 280–297. Springer, 2008.

[157] Frank Kelly. Road Pricing: addressing congestion, pollution and the financing of Britain's roads. *Ingenia*, 29:34–40, 2006.

[158] Dogan Kesdogan, Dakshi Agrawal, and Stefan Penz. Limits of anonymity in open environments. In Fabien A. P. Petitcolas, editor, *5th International Workshop on Information Hiding (IH 2002)*, volume 2578 of *Lecture Notes in Computer Science*, pages 53–69, 2002.

[159] Dogan Kesdogan, Jan Egner, and Roland Büschkes. Stop-and-go-mixes providing probabilistic anonymity in an open system. In David Aucsmith, editor, *2nd International Workshop on Information Hiding*, volume 1525 of *Lecture Notes in Computer Science*, pages 83–98. Springer, 1998.

[160] Dogan Kesdogan, Daniel Mölle, Stefan Richter, and Peter Rossmanith. Breaking anonymity by learning a unique minimum hitting set. In Anna E. Frid, Andrey Morozov, Andrey Rybalchenko, and Klaus W. Wagner, editors, *4th International Computer Science Symposium in Russia (CSR 2009)*, volume 5675 of *Lecture Notes in Computer Science*, pages 299–309. Springer, 2009.

[161] Dogan Kesdogan and Lexi Pimenidis. The hitting set attack on anonymity protocols. In Jessica J. Fridrich, editor, *6th International Workshop on Information Hiding (IH 2004)*, volume 3200 of *Lecture Notes in Computer Science*, pages 326–339. Springer, 2004.

[162] Ali Khoshgozaran and Cyrus Shahabi. Private information retrieval techniques for enabling location privacy in location-based services. In Claudio Bettini, Sushil Jajodia, Pierangela Samarati, and Xiaoyang Sean Wang, editors, *Privacy in Location-Based Applications, Research Issues and Emerging Trends*, volume 5599 of *Lecture Notes in Computer Science*, pages 59–83. Springer, 2009.

[163] Hidetoshi Kido, Yutaka Yanagisawa, and Tetsuji Satoh. An anonymous communication technique using dummies for location-based services. In *International Conference on Pervasive Services (ICPS 2005)*, pages 88–97. IEEE Computer Society, 2005.

[164] Varad Kirtane and C. Pandu Rangan. RSA-TBOS signcryption with proxy re-encryption. In Gregory L. Heileman and Marc Joye, editors, *8th ACM Workshop on Digital Rights Management*, pages 59–66. ACM, 2008.

[165] Negar Kiyavash, Amir Houmansadr, and Nikita Borisov. Multi-flow attack resistant watermarks for network flows. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2009)*, pages 1497–1500. IEEE, 2009.

[166] Markulf Kohlweiss, Sebastian Faust, Lothar Fritsch, Bartek Gedrojc, and Bart Preneel. Efficient oblivious augmented maps: Location-based services with a payment broker. In Nikita Borisov and Philippe Golle, editors, *7th International Symposium on Privacy Enhancing Technologies (PETS 2007)*, volume 4776 of *Lecture Notes in Computer Science*, pages 77–94. Springer, 2007.

[167] John Krumm. Inference attacks on location tracks. In Anthony LaMarca, Marc Langheinrich, and Khai N. Truong, editors, *Pervasive Computing, 5th International Conference*, volume 4480 of *Lecture Notes in Computer Science*, pages 127–143. Springer, 2007.

[168] John Krumm. Realistic driving trips for location privacy. In Hideyuki Tokuda, Michael Beigl, Adrian Friday, A. J. Bernheim Brush, and Yoshito Tobe, editors, *7th International Conference on Pervasive Computing (Pervasive 2009)*, volume 5538 of *Lecture Notes in Computer Science*, pages 25–41. Springer, 2009.

[169] Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[170] Ravi Kumar, Jasmine Novak, and Andrew Tomkins. Structure and evolution of online social networks. In Tina Eliassi-Rad, Lyle H. Ungar, Mark Craven, and Dimitrios Gunopulos, editors, *12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 611–617. ACM, 2006.

[171] Eyal Kushilevitz and Rafail Ostrovsky. Replication is not needed: Single database, computationally-private information retrieval. In *38th Annual Symposium on Foundations of Computer Science (FOCS 97)*, pages 364–373, 1997.

[172] Wenke Lee and Salvatore J. Stolfo. Data mining approaches for intrusion detection. In *7th USENIX Security Symposium*, pages 79–93. USENIX Association, 1998.

[173] Brian N. Levine, Michael K. Reiter, Chenxi Wang, and Matthew K. Wright. Timing attacks in low-latency mix-based systems. In Ari Juels, editor, *Proceedings of Financial Cryptography (FC '04)*, volume 3110 of *Lecture Notes in Computer Science*, pages 251–265. Springer, 2004.

[174] Marc Liberatore and Brian Neil Levine. Inferring the source of encrypted http connections. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM Conference on Computer and Communications Security (CCS 2006)*, pages 255–263. ACM, 2006.

[175] Todd Litman. Distance-based vehicle insurance feasibility, costs and benefits. Technical report, Victoria Transport Policy Institute, 2007.

[176] Jing Liu, Hong yun Xu, and Cheng Xie. A new statistical hitting set attack on anonymity protocols. In *Computational Intelligence and Security, International Conference (CIS 07)*, pages 922–925. IEEE Computer Society, 2007.

[177] VaultLogix LLC. ESRB owns to accidental leak of more than 1 000 email addresses. `http://www.dataprotection.com/online-backup-news/data-protection/esrb-owns-to-accidental-leak-of-more-than-1000-email-addresses-29274`, July 2010. Accessed March 2011.

[178] ARM Ltd. MCB2300 Evaluation Board Family. `http://www.keil.com/mcb2300/`. Accessed March 2011.

[179] David Lyon. Editorial. Surveillance Studies: Understanding visibility, mobility and the phenetic fix. *Surveillance and Society*, 1(1), 2002.

[180] David J. C. Mackay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[181] Nayantara Mallesh and Matthew Wright. The reverse statistical disclosure attack. In Rainer Böhme, Philip W. L. Fong, and Reihaneh Safavi-Naini, editors, *Information Hiding - 12th International Conference (IH 2010)*, volume 6387 of *Lecture Notes in Computer Science*, pages 221–234. Springer, 2010.

[182] MAPFRE. Ycar. `http://www.jovenesdesiguales.com/`. Accessed March 2011.

[183] Nick Mathewson and Roger Dingledine. Practical traffic analysis: Extending and resisting statistical disclosure. In David Martin and Andrei Serjantov, editors, *4th International Workshop on Privacy Enhancing Technologies (PET 2004)*, volume 3424 of *Lecture Notes in Computer Science*, pages 17–34. Springer, 2004.

[184] Jon McLachlan and Nicholas Hopper. Don't clog the queue! circuit clogging and mitigation in P2P anonymity schemes. In Gene Tsudik, editor, *12th International Conference on Financial Cryptography and Data Security (FC 2008)*, volume 5143 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2008.

[185] Jon McLachlan, Andrew Tran, Nicholas Hopper, and Yongdae Kim. Scalable onion routing with Torsk. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *ACM Conference on Computer and Communications Security*, pages 590–599. ACM, 2009.

[186] Alfred Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, 1996.

[187] Alan Mislove, Gaurav Oberoi, Ansley Post, Charles Reis, Peter Druschel, and Dan S. Wallach. AP3: cooperative, decentralized anonymous communication. In Yolande Berbers and Miguel Castro, editors, *ACM SIGOPS European Workshop*, page 30. ACM, 2004.

[188] Prateek Mittal and Nikita Borisov. Information leaks in structured peer-to-peer anonymous communication systems. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *ACM Conference on Computer and Communications Security (CCS 2008)*, pages 267–278. ACM, 2008.

[189] Prateek Mittal and Nikita Borisov. ShadowWalker: peer-to-peer anonymous communication using redundant structured topologies. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *ACM Conference on Computer and Communications Security*, pages 161–172. ACM, 2009.

[190] Prateek Mittal, Nikita Borisov, Carmela Troncoso, and Alfredo Rial. Scalable anonymous communication with provable security. In *5th USENIX Workshop on Hot Topics in Security (HotSec 2010)*, page 7. USENIX, 2010.

[191] Mohamed F. Mokbel, Chi-Yin Chow, and Walid G. Aref. The new Casper: A privacy-aware location-based database server. In *23rd International Conference on Data Engineering (ICDE)*, pages 1499–1500. IEEE, 2007.

[192] Ulf Möller, Lance Cottrell, Peter Palfrader, and Len Sassaman. Mixmaster Protocol — Version 2. IETF Internet Draft, July 2003.

[193] Juan Pedro Muñoz-Gea, Josemaria Malgosa-Sanahuja, Pilar Manzanares-Lopez, Juan Carlos Sanchez-Aarnoutse, and Joan García-Haro. A low-variance random-walk procedure to provide anonymity in overlay networks. In Sushil Jajodia and Javier López, editors, *13th European Symposium on Research in Computer Security (ESORICS 2008)*, volume 5283 of *Lecture Notes in Computer Science*, pages 238–250. Springer, 2008.

[194] Steven J. Murdoch and George Danezis. Low-cost traffic analysis of Tor. In *IEEE Symposium on Security and Privacy (S&P 2005)*, pages 183–195. IEEE Computer Society, 2005.

[195] Steven J. Murdoch and Robert N. M. Watson. Metrics for security and performance in low-latency anonymity systems. In Nikita Borisov and Ian Goldberg, editors, *Privacy Enhancing Technologies Symposium (PETS 2008)*, volume 5134 of *Lecture Notes in Computer Science*, pages 115–132. Springer, 2008.

[196] Steven J. Murdoch and Piotr Zielinski. Sampled traffic analysis by internet-exchange-level adversaries. In Nikita Borisov and Philippe Golle, editors, *7th International Symposium on Privacy Enhancing Technologies (PETS 2007)*, volume 4776 of *Lecture Notes in Computer Science*, pages 167–183. Springer, 2007.

[197] David Naccache and David M'Raihi. Cryptographic smart cards. *IEEE Micro*, 16(3):14–24, 1996.

[198] Shishir Nagaraja, Prateek Mittal, Chi yao Hong, Matthew Caesar, and Nikita Borisov. BotGrep: Finding P2P bots with structured graph analysis. In *19th USENIX Security Symposium*, pages 95 – 110. USENIX Association, 2010.

[199] Arjun Nambiar and Matthew Wright. Salsa: a structured approach to large-scale anonymity. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *13th ACM Conference on Computer and Communications Security (CCS 2006)*, pages 17–26. ACM, 2006.

[200] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy (S&P 2008)*, pages 111–125. IEEE Computer Society, 2008.

[201] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *IEEE Symposium on Security and Privacy (S&P 2009)*, pages 173–187. IEEE Computer Society, 2009.

[202] Arvind Narayanan and Vitaly Shmatikov. Myths and fallacies of "personally identifiable information". *Communications of the ACM*, 53(6):24–26, 2010.

[203] Helen Nissenbaum. Privacy as contextual integrity. *Washington Law Review*, 79(1), 2004.

[204] NIST. *Advanced Encryption Standard (AES) (FIPS PUB 197).* National Institute of Standards and Technology, November 2001.

[205] NXP Semiconductors. LPC23xx User Manual. `http://ics.nxp.com/`. Accessed March 2011.

[206] NXP Semiconductors. SmartMX P5xC012/020/024/ 037/052 family. Secure contact PKI smart card controller.

[207] Luke O'Connor. On blending attacks for mixes with memory. In Mauro Barni, Jordi Herrera-Joancomartí, Stefan Katzenbeisser, and Fernando Pérez-González, editors, *7th International Workshop on Information Hiding (IH05)*, volume 3727 of *Lecture Notes in Computer Science*, pages 39–52. Springer, 2005.

[208] Octo Telematics S.p.A. `http://www.octotelematics.com/`. Accessed March 2011.

[209] Paul Ohm. Broken promises of privacy: Responding to the surprising failure of anonymization. Technical report, University of Colorado Law School, 2009.

[210] Femi G. Olumofin, Piotr K. Tysowski, Ian Goldberg, and Urs Hengartner. Achieving efficient query privacy for location based services. In Mikhail J. Atallah and Nicholas J. Hopper, editors, *10th International Symposium on Privacy Enhancing Technologies (PETS 2010)*, volume 6205 of *Lecture Notes in Computer Science*, pages 93–110. Springer, 2010.

[211] National Committee on Uniform Traffic Laws and Ordinances. Automated traffic law enforcement model law, 2004.

[212] Lasse Øverlier and Paul Syverson. Locating hidden servers. In *Proceedings of the 2006 IEEE Symposium on Security and Privacy*. IEEE Computer Society, May 2006.

[213] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.

[214] Andriy Panchenko and Lexi Pimenidis. Towards practical attacker classification for risk analysis in anonymous communication. In Herbert Leitold and Evangelos P. Markatos, editors, *10th IFIP Communications and Multimedia Security (CMS 2006)*, volume 4237 of *Lecture Notes in Computer Science*, pages 240–251. Springer, 2006.

[215] Andriy Panchenko, Stefan Richter, and Arne Rache. NISAN: network information service for anonymization networks. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *ACM Conference on Computer and Communications Security (CCS 2009)*, pages 141–150. ACM, 2009.

[216] Moon-Hee Park, Jin-Hyuk Hong, and Sung-Bae Cho. Location-based recommendation system using Bayesian user's preference model in mobile devices. In Jadwiga Indulska, Jianhua Ma, Laurence Tianruo Yang, Theo Ungerer, and Jiannong Cao, editors, *4th International Conference on Ubiquitous Intelligence and Computing (UIC2007)*, volume 4611 of *Lecture Notes in Computer Science*, pages 1130–1139. Springer, 2007.

[217] Andreas Pfitzmann and Marit Hansen. Anonymity, unlinkability, unobservability, pseudonymity, and identity management: a consolidated proposal for terminology. Technical report, TU Dresden, 2008.

[218] Birgit Pfitzmann. Breaking efficient anonymous channel. In *EUROCRYPT*, pages 332–340, 1994.

[219] B. J. Phillips, C. D. Schmidt, and D. R. Kelly. Recovering data from USB flash memory sticks that have been damaged or electronically erased. In *Proceedings of the 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia and workshop*, e-Forensics '08, pages 19:1–19:6. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[220] Raluca Popa, Hari Balakrishnan, and Andrew Blumberg. VPriv: Protecting privacy in location-based vehicular services. In *18th Usenix Security Symposium*, pages 335–350. USENIX Association, 2009.

[221] Young June Pyun, Young Hee Park, Xinyuan Wang, Douglas S. Reeves, and Peng Ning. Tracing traffic through intermediate hosts that repacketize flows. In *26th IEEE International Conference on Computer Communications (INFOCOM 2007)*, pages 634–642. IEEE, 2007.

[222] Michael O. Rabin. How to exchange secrets with oblivious transfer. Technical report, Harvard Aiken Computation Laboratory, 1981. `http://eprint.iacr.org/2005/187`.

[223] Stefan Rass, Simone Fuchs, Martin Schaffer, and Kyandoghere Kyamakya. How to protect privacy in floating car data systems. In Varsha K. Sadekar, Paolo Santi, Yih-Chun Hu, and Martin Mauve, editors, *5th International Workshop on Vehicular Ad Hoc Networks (VANET 2008)*, pages 17–22. ACM, 2008.

[224] Jean-François Raymond. Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In H. Federrath, editor, *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, volume 2009 of *Lecture Notes in Computer Science*, pages 10–29. Springer-Verlag, July 2000.

[225] Jean-Charles Régin. A filtering algorithm for constraints of difference in CSPs. In *12th National Conference on Artificial Intelligence (AAAI)*, pages 362–367, 1994.

[226] Michael Reiter and Aviel Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.

[227] Marc Rennhard and Bernhard Plattner. Introducing MorphMix: Peer-to-Peer based Anonymous Internet Usage with Collusion Detection. In Sushil Jajodia and Pierangela Samarati, editors, *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2002)*, pages 91–102. ACM, 2002.

[228] Alfredo Rial and George Danezis. Privacy-preserving smart metering. Microsoft technical report MSR-TR-2010-150, November 2010.

[229] Phillip Rogaway and Thomas Shrimpton. A provable-security treatment of the key-wrap problem, 2006.

[230] Kazue Sako and Joe Kilian. Receipt-free mix-type voting scheme - a practical solution to the implementation of a voting booth. In Louis C. Guillou and Jean-Jacques Quisquater, editors, *International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT 95)*, volume 921 of *Lecture Notes in Computer Science*, pages 393–403. Springer, 1995.

[231] Pierangela Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[232] Pierangela Samarati and Latanya Sweeney. Generalizing data to provide anonymity when disclosing information (abstract). In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 98)*, page 188. ACM Press, 1998.

[233] Krishna Sampigethaya, Leping Huang, Mingyan Li, Radha Poovendran, Kanta Matsuura, and Kaoru Sezaki. CARAVAN: Providing location privacy for vanet. In Jean-Pierre Hubaux and Christof Paar, editors, *Embedded Security in Cars (escar 2005)*, 2005.

[234] Krishna Sampigethaya, Mingyan Li, Leping Huang, and Radha Poovendran. AMOEBA: Robust Location Privacy Scheme for VANET. *IEEE Journal on Selected Areas in Communications*, 25(8):1569–1589, 2007.

[235] Jochen H. Schiller and Agnès Voisard, editors. *Location-Based Services.* Morgan Kaufmann, 2004.

[236] Max Schuchard, Alexander W. Dean, Victor Heorhiadi, Nicholas Hopper, and Yongdae Kim. Balancing the shadows. In Keith B. Frikken, editor, *ACM Workshop on Privacy in the Electronic Society (WPES 2010)*, pages 1–10. ACM, 2010.

[237] Andrei Serjantov. *On the Anonymity of Anonymity Systems.* PhD thesis, University of Cambridge, June 2004.

[238] Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In Roger Dingledine and Paul F. Syverson, editors, *2nd International Workshop on Privacy Enhancing Technologies (PET 2002)*, volume 2482 of *Lecture Notes in Computer Science*, pages 41–53. Springer, 2002.

[239] Andrei Serjantov, Roger Dingledine, and Paul Syverson. From a trickle to a flood: Active attacks on several mix types. In Fabien Petitcolas, editor, *Proceedings of Information Hiding Workshop (IH 2002).* Springer-Verlag, Lecture Notes in Computer Science 2578, October 2002.

[240] Andrei Serjantov and Richard E. Newman. On the anonymity of timed pool mixes. In Dimitris Gritzalis, Sabrina De Capitani di Vimercati, Pierangela Samarati, and Sokratis K. Katsikas, editors, *18th International Conference on Information Security (SEC2003). Security and Privacy in the Age of Uncertainty, IFIP TC11*, volume 250 of *IFIP Conference Proceedings*, pages 427–434. Kluwer, 2003.

[241] Andrei Serjantov and Peter Sewell. Passive attack analysis for connection-based anonymity systems. In Einar Snekkenes and Dieter Gollmann, editors, *8th European Symposium on Research in Computer Security (ESORICS 2003)*, volume 2808 of *Lecture Notes in Computer Science*, pages 116–131. Springer, 2003.

[242] Andrei Serjantov and Peter Sewell. Passive-attack analysis for connection-based anonymity systems. *International Journal on Information Security*, 4(3):172–180, 2005.

[243] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[244] Claude Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423:623–656, 1948.

[245] Erik Shimshock, Matt Staats, and Nick Hopper. Breaking and provably fixing minx. In Nikita Borisov and Ian Goldberg, editors, *8th Privacy Enhancing Technologies SymposiumPETS 2008 ()*, volume 5134 of *Lecture Notes in Computer Science*, pages 99–114. Springer, 2008.

[246] Vitaly Shmatikov. Probabilistic analysis of an anonymity system. *Journal of Computer Security*, 12(3-4):355–377, 2004.

[247] Reza Shokri, Julien Freudiger, and Jean-Pierre Hubaux. A unified framework for location privacy. Technical report, LCA, EPFL, Switzerland, 2010.

[248] Reza Shokri, Carmela Troncoso, Claudia Diaz, Julien Freudiger, and Jean-Pierre Hubaux. Unraveling an old cloak: k-anonymity for location privacy. In Keith Frikken, editor, *9th ACM workshop on Privacy in the electronic society (WPES 2010)*, pages 115–118. ACM, 2010.

[249] Gustavus J. Simmons. Subliminal communication is easy using the DSA. In Tor Helleseth, editor, *Workshop on the Theory and Application of of Cryptographic Techniques (EUROCRYPT 1993)*, volume 765 of *Lecture Notes in Computer Science*, pages 218–232. Springer, 1993.

[250] Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The Annals of Mathematical Statistics*, 35(2):876–879, 1964.

[251] Christopher Soghoian. 8 million reasons for real surveillance oversight. `http://paranoia.dubfire.net/2009/12/8-million-reasons-for-real-surveillance.html`, December 2009. Accessed March 2011.

[252] Christopher Soghoian. An end to privacy theater: Exposing and discouraging corporate disclosure of user data to the government. *Minnesota Journal of Law, Science & Technology, Forthcoming*, 2010.

[253] Daniel J. Solove. A taxonomy of privacy. *University of Pennsylvania Law Review*, 154(3):477, 2006.

[254] Joo-Han Song, Vincent W. S. Wong, and Victor C. M. Leung. Wireless location privacy protection in vehicular ad-hoc networks. In *IEEE International Conference on Communications (ICC 2009)*, pages 1–6. IEEE, 2009.

[255] Joo-Han Song, Vincent W. S. Wong, and Victor C. M. Leung. Wireless location privacy protection in vehicular ad-hoc networks. *Mobile Network Applications*, 15(1):160–171, 2010.

[256] STOK. STOK: Telematics and pay as you drive solutions. `http://www.stok-nederland.nl/`. Accessed March 2011.

[257] Latanya Sweeney. k-anonymity: a model for protecting privacy. *International journal of uncertainty fuzziness and knowledge-based systems*, 10(5):557–570, 2002.

[258] Paul F. Syverson, Michael G. Reed, and David M. Goldschlag. Private web browsing. *Journal of Computer Security*, 5(3):237–248, 1997.

[259] Paul F. Syverson, Gene Tsudik, Michael G. Reed, and Carl E. Landwehr. Towards an analysis of onion routing security. In Hannes Federrath, editor, *Designing Privacy Enhancing Technologies, Internationa Workshop on Design Issues in Anonymity and Unobservability,*, volume 2009 of *Lecture Notes in Computer Science*, pages 96–114. Springer, 2000.

[260] Parisa Tabriz and Nikita Borisov. Breaking the collusion detection mechanism of morphmix. In George Danezis and Philippe Golle, editors, *Privacy Enhancing Technologies (PET 2006)*, volume 4258 of *Lecture Notes in Computer Science*, pages 368–383. Springer, 2006.

[261] Kar Way Tan, Yimin Lin, and Kyriakos Mouratidis. Spatial cloaking revisited: Distinguishing information leakage from anonymity. In Nikos Mamoulis, Thomas Seidl, Torben Bach Pedersen, Kristian Torp, and Ira Assent, editors, *11th International Symposium on Advances in Spatial and Temporal Databases (SSTD 2009)*, volume 5644 of *Lecture Notes in Computer Science*, pages 117–134. Springer, 2009.

[262] Telit. GM862-GPS Hardware User Guide. `www.telit.com/module/infopool/download.php?id=537`. Accessed March 2011.

[263] Gergely Tóth and Zoltán Hornák. Measuring anonymity in a non-adaptive, real-time system. In David Martin and Andrei Serjantov, editors, *4th International Workshop on Privacy Enhancing Technologies (PET 2004)*, volume 3424 of *Lecture Notes in Computer Science*, pages 226–241. Springer, 2004.

[264] Trafficmaster. Real Time and Historic Data feeds. `http://www.trafficmaster.co.uk/content/1/82/real-time-and-historic-data-feeds.html`. Accessed March 2011.

[265] Andrew Tran, Nicholas Hopper, and Yongdae Kim. Hashing it out in public: common failure modes of dht-based anonymity schemes. In Stefano Paraboschi and Ehab Al-Shaer, editors, *8th ACM Workshop on Privacy in the Electronic Society (WPES '09)*, pages 71–80. ACM, 2009.

[266] Carmela Troncoso and George Danezis. The Bayesian traffic analysis of mix networks. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *Conference on Computer and Communications Security (CCS 2009)*, pages 369–379. ACM, 2009.

[267] Carmela Troncoso, George Danezis, Eleni Kosta, Josep Balasch, and Bart Preneel. PriPAYD: Privacy Friendly Pay-As-You-Drive Insurance. *IEEE Transactions on Dependable and Secure Computing*, (To appear), 2011.

[268] Carmela Troncoso, George Danezis, Eleni Kosta, and Bart Preneel. PriPAYD: privacy friendly pay-as-you-drive insurance. In Peng Ning and Ting Yu, editors, *ACM Workshop on Privacy in the Electronic Society (WPES 2007)*, pages 99–107. ACM, 2007.

[269] Carmela Troncoso, Claudia Diaz, Orr Dunkelman, and Bart Preneel. Traffic analysis attacks on a continuously-observable steganographic file system. In Teddy Furon, François Cayre, Gwenaël J. Doërr, and Patrick Bas, editors, *9th International Workshop on Information Hiding (IH 2007)*, volume 4567 of *Lecture Notes in Computer Science*, pages 220–236. Springer, 2007.

[270] Carmela Troncoso, Benedikt Gierlichs, Bart Preneel, and Ingrid Verbauwhede. Perfect matching disclosure attacks. In Nikita Borisov and Ian Goldberg, editors, *8th International Symposium on Privacy Enhancing Technologies (PETS 2008)*, volume 5134 of *Lecture Notes in Computer Science*, pages 2–23. Springer-Verlag, 2008.

[271] Hung-Wen Tung and Von-Wun Soo. A personalized restaurant recommender agent for mobile e-service. In *International Conference on e-Technology, e-Commerce, and e-Services (EEE 04)*, pages 259–262. IEEE Computer Society, 2004.

[272] Uniqa. `http://www.uniqa.at/uniqa_at/`. Accessed March 2011.

[273] US Department Of Transport. Intellidrive program: Vehicle to vehicle safety application research plan. `http://www.fmcsa.dot.gov/facts-research/media/webinar-10-01-20-slides.pdf`, January 2010. Accessed March 2011.

[274] Gilles W. van Blarkom, John J. Borking, and J.G. Eddy Olk, editors. *Handbook of Privacy and Privacy-Enhancing Technologies: The Case of Intelligent Software Agents*. College Bescherming Persoonsgegevens, 2003.

[275] Qiyan Wang, Prateek Mittal, and Nikita Borisov. In search of an anonymous and secure lookup: attacks on structured peer-to-peer anonymous communication systems. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *ACM Conference on Computer and Communications Security (CCS 2010)*, pages 308–318. ACM, 2010.

[276] Xinyuan Wang, Shiping Chen, and Sushil Jajodia. Tracking anonymous peer-to-peer voip calls on the internet. In Vijay Atluri, Catherine Meadows, and Ari Juels, editors, *ACM Conference on Computer and Communications Security (CCS 2005)*, pages 81–91. ACM, 2005.

[277] Xinyuan Wang, Shiping Chen, and Sushil Jajodia. Network flow watermarking attack on low-latency anonymous communication systems. In *IEEE Symposium on Security and Privacy (S&P 2007)*, pages 116–130. IEEE Computer Society, 2007.

[278] Xinyuan Wang and Douglas S. Reeves. Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays. In Sushil Jajodia, Vijayalakshmi Atluri, and Trent Jaeger, editors, *ACM Conference on Computer and Communications Security (CCS 2003)*, pages 20–29. ACM, 2003.

[279] Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications (Structural Analysis in the Social Sciences)*. Cambridge University Press, 1994.

[280] Benedikt Westermann, Rolf Wendolsky, Lexi Pimenidis, and Dogan Kesdogan. Cryptographic protocol analysis of an.on. In Radu Sion, editor, *14th International Conference Financial Cryptography and Data Security (FC 2010)*, volume 6052 of *Lecture Notes in Computer Science*, pages 114–128. Springer, 2010.

[281] Christo Wilson, Bryce Boe, Alessandra Sala, Krishna P. N. Puttaswamy, and Ben Y. Zhao. User interactions in social networks and their implications. In Wolfgang Schröder-Preikschat, John Wilkes, and Rebecca Isaacs, editors, *EuroSys Conference 2009*, pages 205–218. ACM, 2009.

[282] Matthew Wright, Micah Adler, Brian Neil Levine, and Clay Shields. An analysis of the degradation of anonymous protocols. In *Proceedings of the Network and Distributed Security Symposium - NDSS '02*, pages 21–33. IEEE, 2002.

[283] Matthew Wright, Micah Adler, Brian Neil Levine, and Clay Shields. The predecessor attack: An analysis of a threat to anonymous communications systems. *ACM Transactions on Information and System Security*, 4(7):489–522, 2004.

[284] Toby Xu and Ying Cai. Feeling-based location privacy protection for location-based services. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *ACM Conference on Computer and Communications Security (CCS 2009)*, pages 348–357. ACM, 2009.

[285] Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science (FOCS 1982)*, pages 160–164. IEEE, 1982.

[286] Tun-Hao You, Wen-Chih Peng, and Wang-Chien Lee. Protecting moving trajectories with dummies. In Christian Becker, Christian S. Jensen, Jianwen Su, and Daniela Nicklas, editors, *8th International Conference on Mobile Data Management (MDM 2007)*, pages 278–282. IEEE, 2007.

[287] Fan Yu and Subir K. Biswas. Impacts of radio access protocols on cooperative vehicle collision avoidance in urban traffic intersections. *Journal of Communication*, 3(4):41–48, 2008.

[288] Haifeng Yu, Phillip B. Gibbons, Michael Kaminsky, and Feng Xiao. SybilLimit: A near-optimal social network defense against Sybil attacks. In *IEEE Symposium on Security and Privacy (S&P 2008)*, pages 3–17. IEEE Computer Society, 2008.

[289] Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham D. Flaxman. SybilGuard: defending against Sybil attacks via social networks. *IEEE/ACM Transactions on Networking*, 16(3):576–589, 2008.

[290] Wei Yu, Xinwen Fu, Steve Graham, Dong Xuan, and Wei Zhao. DSSS-based flow marking technique for invisible traceback. In *IEEE Symposium on Security and Privacy (S&P 2007)*, pages 18–32. IEEE Computer Society, 2007.

[291] Fayyaz Zahid and Craig Barton. Pay per mile insurance. Technical report, Davenport University, 2004.

[292] Ge Zhong, Ian Goldberg, and Urs Hengartner. Louis, Lester and Pierre: Three protocols for location privacy. In Nikita Borisov and Philippe Golle,

editors, *7th International Symposium on Privacy Enhancing Technologies (PETS 2007)*, volume 4776 of *Lecture Notes in Computer Science*, pages 62–76. Springer, 2007.

[293] Ge Zhong and Urs Hengartner. A distributed k-anonymity protocol for location privacy. In *IEEE International Conference on Pervasive Computing and Communications (PerCom 2009)*, pages 1–10. IEEE Computer Society, 2009.

[294] Xuan Zhou, HweeHwa Pang, and Kian-Lee Tan. Hiding data accesses in steganographic file system. In *Proceedings of the 20th International Conference on Data Engineering*, pages 572–583. IEEE Computer Society, 2004.

List of Publications

International Journals

2011 C. Troncoso, E. Costa-Montenegro, C. Diaz, and S. Schiffner. How the Vehicle Infrastructure Integration anonymous certificates enable the tracking and re-identification of vehicles. In *Journal on Computer Networks: Special Issue on Vehicle-2-x Communication*, 26 pages, Accepted 2011.

C. Troncoso, G. Danezis, E. Kosta, J. Balasch, and B. Preneel. PriPAYD: Privacy Friendly Pay-As-You-Drive Insurance. In *IEEE Transactions on Dependable and Secure Computing*, 14 pages, Submitted 2009, Accepted 2010.

International Conferences

2011 S. F. Gürses, C. Troncoso, and C. Diaz. Engineering Privacy by Design. In *Computers, Privacy & Data Protection (CPDP 2011)*, 25 pages, 2011.

2010 J. Balasch, A. Rial, C. Troncoso, C. Geuens, B. Preneel, and I. Verbauwhede. PrETP: Privacy-Preserving Electronic Toll Pricing. In *19th USENIX Security Symposium 2010*, USENIX, pp. 63-78, 2010.

G. Danezis, C. Diaz, C. Troncoso, and B. Laurie. Drac: An Architecture for Anonymous Low-Volume Communications. In *10th International Symposium on Privacy Enhancing Technologies, PETS 2010*, LNCS 6205, M. J. Atallah, and N. J. Hopper (eds.), Springer-Verlag, pp. 202-219, 2010.

2010 C. Diaz, S. Murdoch, and C. Troncoso. Impact of Network Topology on Anonymity and Overhead in Low-Latency Anonymity Networks. In *10th International Symposium on Privacy Enhancing Technologies, PETS 2010*, LNCS 6205, M. J. Atallah, and N. J. Hopper (eds.), Springer-Verlag, pp. 184-201, 2010.

P. Mittal, N. Borisov, A. Rial, and C. Troncoso. Scalable Anonymous Communication with Provable Security. In *5th USENIX Workshop on Hot Topics in Security 2010*, USENIX, 7 pages, 2010.

R. Shokri, C. Troncoso, C. Diaz, J. Freudiger, and J. Hubaux. Unraveling an Old Cloak: k-anonymity for Location Privacy. In *Proceedings of the 9th ACM Workshop on Privacy in the Electronic Society (WPES 2010)*, K. Frikken (ed.), ACM, pp. 115-118, 2010.

2009 G. Danezis, C. Diaz, E. Käsper, and C. Troncoso. The wisdom of Crowds: attacks and optimal constructions. In *14th European Symposium on Research in Computer Security (ESORICS 2009)*, LNCS 5789, M. Backes, and P. Ning (eds.), Springer-Verlag, pp. 406-423, 2009.

G. Danezis, and C. Troncoso. Vida: How to use Bayesian inference to de-anonymize persistent communications. In *9th International Symposium on Privacy Enhancing Technologies, PETS 2009*, LNCS 5672, M. J. Atallah, and I. Goldberg (eds.), Springer-Verlag, pp. 406-423, 2009.

C. Troncoso, and G. Danezis. The Bayesian Analysis of Mix Networks. In *16th ACM Conference on Computer and Communications Security (CCS 2009)*, E. Al-Shaer, S. Jha, and A. D. Keromytis (eds.), ACM, pp. 369-379, 2009.

2008 C. Diaz, C. Troncoso, and B. Preneel. A Framework for the Analysis of Mix-Based Steganographic File Systems. In *13th European Symposium on Research in Computer Security (ESORICS 2008)*, LNCS 5283, S. Jajodia, and J. Lopez (eds.), Springer-Verlag, pp. 428-445, 2008.

C. Diaz, C. Troncoso, and A. Serjantov. On the Impact of Social Network Profiling on Anonymity. In *8th International Symposium on Privacy Enhancing Technologies, PETS 2008*, LNCS 5134, N. Borisov, and I. Goldberg (eds.), Springer-Verlag, pp. 44-62, 2008.

B. Gierlichs, C. Troncoso, C. Diaz, B. Preneel, and I. Verbauwhede. Revisiting A Combinatorial Approach Toward Measuring Anonymity. In *7th ACM Workshop on Privacy in the Electronic Society (WPES 2008)*, V. Atluri , and M. Winslett (eds.), ACM, pp. 111-116, 2008.

2008 C. Troncoso, B. Gierlichs, B. Preneel, and I. Verbauwhede. Perfect Matching Disclosure Attacks. In *8th International Symposium on Privacy Enhancing Technologies, PETS 2008*, LNCS 5134, N. Borisov, and I. Goldberg (eds.), Springer-Verlag, pp. 2-23, 2008.

C. Troncoso, D. De Cock, and B. Preneel. Improving Secure Long-Term Archival of Digitally Signed Documents. In *4th International Workshop on Storage Security and Survivability (StorageSS 2008)*, Y. Kim, and B. Yurcik (eds.), pp. 27-36, 2008.

2007 G. Danezis, C. Diaz, S. Faust, E. Käsper, C. Troncoso, and B. Preneel. Efficient Negative Databases from Cryptographic Hash Functions. In *10th International Conference on Information Security, ISC 2007*, LNCS 4779, J. A. Garay, A. K. Lenstra, M. Mambo, and R. Peralta (eds.), Springer-Verlag, pp. 423-436, 2007.

G. Danezis, C. Diaz, and C. Troncoso. Two-Sided Statistical Disclosure Attack. In *7th International Symposium on Privacy Enhancing Technologies, PETS 2007*, N. Borisov and P. Golle (Eds), Springer LNCS 4776, pp. 30-44, 2007.

C. Diaz, C. Troncoso, and G. Danezis. Does additional information always reduce anonymity? In *6th ACM Workshop on Privacy in the Electronic Society (WPES 2007)*, T. Yu (ed.), ACM, pp. 72-75, 2007.

C. Troncoso, G. Danezis, E. Kosta, and B. Preneel. PriPAYD: Privacy Friendly Pay-As-You-Drive Insurance. In *6th ACM Workshop on Privacy in the Electronic Society (WPES 2007)*, T. Yu (ed.), ACM, pp. 99-107, 2007.

C. Troncoso, C. Diaz, O. Dunkelman, and B. Preneel. Traffic Analysis Attacks on a Continuously-Observable Steganographic File. In *9th International Workshop on Information Hiding, IH 2007*, LNCS 4567, F. Cayre, G. J. Doërr, and T. Furon (eds.), Springer-Verlag, pp. 220-236, 2007.

Curriculum vitae

Carmela Troncoso was born on September 17th 1982 in Vigo, Spain. She received the Master's degree in Telecommunications Engineering (Ingeniero de Telecomunicación) from the University of Vigo, Spain, in May 2006.

She joined the COSIC research group at the Katholieke Universiteit Leuven, Belgium as a PhD student in October 2006. The research for her PhD studies lead to more than 20 publications in peer-reviewed, international conferences and journals, including six publications at the Privacy Enhancing Technologies Symposium (PETS). She has served on more than 10 program committees of international conferences, has been chair of the Hot Topics in Privacy Enhancing Technologies Workshop (HotPETs) in 2010 and 2011, and has reviewed articles for numerous international journals and conferences.

Arenberg Doctoral School of Science, Engineering & Technology

Faculty of Engineering

Department of Electrical Engineering (ESAT)

COmputer Security and Industrial Cryptography (COSIC)

Kasteelpark Arenberg 10, 2446, 3001 Heverlee, Belgium

KATHOLIEKE UNIVERSITEIT LEUVEN