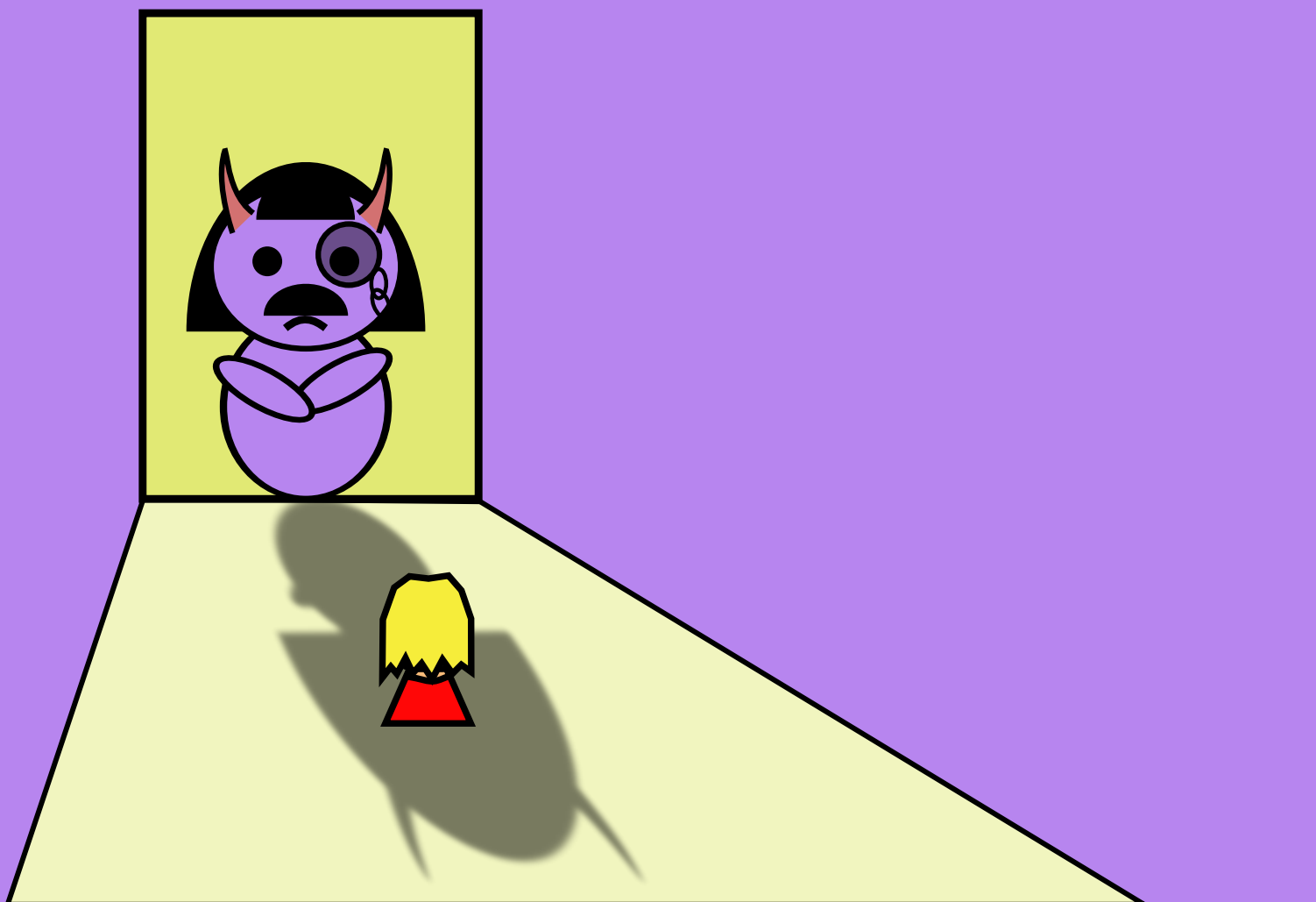
Universidad de Vigo

A STATISTICAL APPROACH TO THE DESIGN OF PRIVACY-PRESERVING SERVICES

Simon Oya | PhD Thesis

Supervised by:
Carmela Troncoso
Fernando Pérez-González



This thesis was partially funded by the Galician Regional Government and Spanish Ministry of Education, Culture and Sport under the FPU grant; by the Galician Regional Government and the European Regional Development Fund (ERDF) under projects Consolidation of Research Units 2009/62, 2010/85, 2013/09; by the Spanish Government and ERDF under projects COMONSENS (CONSOLIDER-INGENIO 2010 CSD2008-00010), COMPASS (TEC2013-47020-C2-1-R), and TACTICA; by the Agencia Estatal de Investigación (Spain) and ERDF under project WINTER (TEC2016-76409-C2-2-R); by atlanTTic and the EU H2020 Framework Programme through the WITDOM Project under Grant 644371; by the EU H2020-ICT-10-2015 NEXTLEAP Program under Grant 688722; by the Xunta de Galicia and ERDF under projects Agrupación Estratéxica Consolidada de Galicia accreditation 2016-2019; by RedTemática RedTEIC 2017-2018 and by the Fundación Barrié under Programa de Becas de Posgrado en el Extranjero.

Abstract

Electronic services have become an indispensable part of society. Billions of users rely on these services every day to communicate with friends, meet new people, buy products, and keep track of their activities. Electronic services provide many comforts to society, but also pose new threats to the privacy of their users. This is due to the fact that users of electronic services send their sensitive information over a communication channel (typically, the Internet), and this information can many times be observed by an unwanted party.

Even though encryption can protect the content of communications against unwanted observers, there are other privacy problems that encryption does not solve. In this thesis, we study two of these problems. First, we tackle the problem of meta-data leakage against a passive eavesdropper. Meta-data is information related to a communication other than the content of the communication itself, such as who the communicating parties are, how often they communicate, or where they are located. Meta-data is usually sensitive, so it is important that users hide it from eavesdroppers. In the first part of this thesis, we study a particular solution to meta-data leakage: mix-based anonymous communication systems. We analyze these systems, and find out how to optimally configure their parameters so as to maximize the users' privacy.

In the second part of the thesis, we study how to protect users against an adversarial service provider. We consider the particular case of Location-Based Services (LBS), where users want to obtain some service that depends on their real location (e.g., finding nearby points of interest), but do not want to share this location with the service provider. We study obfuscation-based location privacy mechanisms, that allow users to obtain some utility from the LBS without revealing their actual location. We find weaknesses in the approach that previous works follow to design and evaluate location privacy-preserving mechanisms, and propose solutions to mend these issues and improve current designs.

Throughout the thesis we follow a *statistical approach* to improve the privacy of electronic services: we model both the system operation and the users' behavior, and leverage these models to optimize the privacy of the systems. This approach provides theoretical guarantees that our results will be universally valid

as long as the models that we assume for user behavior hold. Also, our methodology can be easily adapted to other privacy problems, and we hope it will inspire future research in this direction.

Acknowledgments

I started my PhD almost seven years ago, back in November 2012, and I have come a long way since then. I want to take this opportunity to thank the many people that have inspired me to learn, overcome challenges that I encountered, and thrive as a researcher.

I want to begin by thanking my supervisors, Carmela Troncoso and Fernando Pérez-González. Thank you for your guidance and advice, and for your support during my "drama-queen" moments. You are the main reasons that I started this journey and I couldn't have made it this far without you.

I want to thank jury members Catuscia Palamidessi, Pedro Comesaña Alfaro and Steven J. Murdoch for agreeing to evaluate this work and coming all the way to Vigo for my defense. Further, thanks to Juan Troncoso and Joaquín García Alfaro for reading my thesis and writing the reports that I needed to get the international mention, and to Prof. Anand Sarwate for guiding me during my internship at Rutgers University.

There are a lot of people that have made my life at the university more fulfilling during this project. Miguel, thanks for splitting bocadillos de tortilla with me and for amusing the lab with your mindless debates. Thank you David for being on my side in most of those silly discussions we had at the lab; it felt great to have at least *one* other reasonable person in the lab. To the man that gets overly excited at the anticipation of free food, Alberto Pedrouzo, thank you for keeping the lab awake with your surprise volume shifts. To my fellow PhD students that defended a bit earlier than me, Magui and Fátima: suffering together made the torments more bearable. I also want to thank other people that are not part of this lab, but were for a long time part of our lunch squad and I am glad we still gather for dinner from time to time: Vigas, Paulinha, Rodelgo. Thanks to Gabi for all of the good times we had at the lab.

During these years, we've had many foreign visitors in our lab. I have enjoyed meeting and learning something new from each of them. Special mention to Michael, Serena and Cecilia; it was great sharing some time with you all. Also, I want to thank Omar for the most epic lapsus linguae moment I'll ever witness.

During my internships abroad, I met a lot of people for whom I am grateful. Thank you Zahra, Mohsen, Hafiz, and the others, for being so kind and welcoming during my stay at Rutgers University. Also, thanks to Bogdan, Wouter, and the rest of the people that I met during my internship at EPFL. I'll keep these nine months that I spent abroad very close to my heart. I had the time of my life, and hopefully I will be able to visit again in the future.

There are a lot of people from outside of my academic world without whom I would not have enjoyed this journey as much as I did. My "minigrupo" squad (Tati, Antía, Kesia, Lexy), my other friends from Gondomar (Santi, Mara, Jorge, Sergio), my university friends (Trillo, Rober), and my close friends that live too far away (Jairo, Rubén, Marcel, Maxi): thank you all for reminding me to be happy. Thanks to Benja, who endured many of my past stressful moments, and to Jesse, who has taken them on with his love and support. To my dear American friends (CJ, Richard, Eric) and Thomas from Lausanne: it was great meeting you, and I will see you very soon.

Last but not least, I want to thank those closest to me: my family, especially my parents Belén and Quique. I am truly lucky for having you. Bimba, Lula, China, you're in my heart.

Contents

Acronyms and Abbreviations

EM	Expectation-Maximization
FPR	False Positive Rate
GPS	Global Positioning System
IP	Internet Protocol
ISP	Internet Service Provider
LBS	Location-Based Service
LPPM	Location Privacy-Preserving Mechanism
LSDA	Least Squares Disclosure Attack
MAP	Maximum A-Posteriori
MLE	Maximum Likelihood Estimator
MSE	Mean Squared Error
PEB	Profile Estimation-Based
PMDA	Perfect Matching Disclosure Attack
PoI	Point of Interest
ROC	Receiver Operating Characteristic curve
SDA	Statistical Disclosure Attack
TPR	True Positive Rate
Vida	Bayesian Inference Disclosure Attack

Notation

Throughout this thesis, we use the following notation unless otherwise stated. Upper case characters denote scalar random variables, and lower case characters denote their realizations (e.g., x is a realization of the random variable X). Vectors and matrices are denoted by lower-case and upper-case boldface characters, respectively; whether the values inside them are random variables or realizations will be clear from the context. Sets are denoted using calligraphic letters (e.g., $\mathcal{A}$), and the real plane is denoted by $\mathbb{R}^2$.

We use the following notation for matrix and vector operations. Matrix $\mathbf{A}^T$ is the transpose of $\mathbf{A}$ (same for vectors). We use $\mathrm{Tr}\{\mathbf{A}\}$ to denote the trace of matrix $\mathbf{A}$. Matrix $\mathrm{diag}\{\mathbf{a}\}$ is a diagonal matrix whose main diagonal contains the elements of the vector $\mathbf{a}$. Matrix $\mathbf{I}_{N\times N}$ is the $N \times N$ identity matrix, $\mathbf{1}_{N\times N}$ is the $N \times N$ ones matrix and $\mathbf{1}_\rho$ is the $\rho \times 1$ vector of ones. Likewise, $\mathbf{1}_N$ is the $N \times 1$ column vector of ones. We use $(\mathbf{A})_{m,n}$ to refer to the m,n-th element of matrix $\mathbf{A}$. The Frobenius norm of matrix $\mathbf{A}$ is denoted by $||\mathbf{A}||$. More precisely, if $\mathbf{A}$ is an $M \times N$ matrix and $a_{m,n} = (\mathbf{A})_{m,n}$, then the Frobenius norm is defined as

$$||\mathbf{A}|| \doteq \sqrt{\sum_{m=1}^{M}\sum_{n=1}^{N} a_{m,n}^2}\,. \tag{1}$$

The operator $\circ$ is the entrywise or Hadamard product of matrices, i.e., the elements of matrix $\mathbf{C} = \mathbf{A} \circ \mathbf{B}$ are

$$(\mathbf{C})_{m,n} = (\mathbf{A})_{m,n} \cdot (\mathbf{B})_{m,n} \tag{2}$$

We use $\mathrm{E}\{X\}$ to denote the mathematical expectation of the random variable X, and $\mathrm{E}\{X|Y\}$ to denote the conditional expectation of X given Y. The same applies to vectors and matrices. The entropy of a random varible X is denoted by $H(x)$, and the conditional entropy of X given Y is written as $H(x|y)$.

Finally, we use the circumflex accent to denote an adversary's estimation (e.g., $\hat{\mathbf{A}}$ is the adversary's estimation of $\mathbf{A}$, and the same applies to vectors and scalar values).

Chapter 1

Introduction

In the last decades, we have witnessed the surge of *electronic services*. These services, that rely on communication technologies such as the Internet or mobile phone networks, include applications such as e-mail, instant messaging, online social networks, and location-based services. Today, electronic services are integrated into people's daily activities and it is hard to envision society without the comfort that these services provide. Electronic services require that users send their data over a communication channel, either to interact with other users or with a service provider. This is a privacy problem, since users' transmitted information, as well as their usage patterns (when, where, and how frequently they use a service), can sometimes be observed by unwanted parties.

Take for example the case of an individual, Alice, who uses her phone to browse a website with information about a particular rare disease. She sends a request to the website provider, and gets the webpage with the information she desires in response. During this process, the website provider learns that Alice is interested on learning about that particular rare disease, and she might accept this information disclosure as part of the deal. However, it may be possible that a third party also learns this information: this could be an eavesdropper monitoring the traffic Alice sends and receives (e.g., a wiretapper, or a curious user sniffing Alice wireless connection), but also Alice's Internet service provider. The fact that Alice wants to know more about a rare disease is very sensitive information, and if this information is learned by another party this constitutes a privacy violation. Note that this privacy issue is not particular of web browsing: it is possible to infer sensitive information about a user by analyzing her usage of different electronic services. For example, user check-ins in particular locations can leak the user's sexual orientation, or political and religious beliefs. Protecting this data leakage is a fundamental challenge towards achieving privacy-preserving electronic services.

A solution to protect communication data from observers is to encrypt the

data, i.e., use mathematical techniques to hide the information inside the messages that the user transmits to other parties, so that only the intended recipient can read them. This way, an eavesdropper of the communication that manages to capture messages exchanged between two users cannot read their content, and the users enjoy *communication confidentiality*. However, encryption alone does not solve the privacy issues related to electronic services. First, encryption protects the *content* of the message against unintended recipients, but there is other data, known as meta-data, that encryption does not hide. This meta-data is the information that is part of a communication other than the content of the communication itself, i.e., who communicates with whom, the timing patterns of the communicating parties, their location, the frequency with which they communicate, etc. This information is in many cases highly sensitive, so protecting it from observers is crucial towards achieving privacy-preserving communications. Second, sometimes the user wishes to obtain a service from a particular provider (e.g., download a web page from a web provider) but does not want the service provider itself to know the information being requested. In this case, if the user encryps her messages against the provider, she gets no utility back. Thus, she has to rely on other techniques to reduce the information leakage against this provider.

In this thesis, we study one particular example for each of these scenarios where encryption alone does not protect the privacy of the users of electronic services. In the case meta-data leakage protection, we study mix-based anonymous communication systems that hide the identity of communicating parties. In the case of protection against an adversarial service provider, we study obfuscation-based techniques that provide privacy in Location Based Services (LBS). In both cases, our goal is to optimize the design of these systems to improve the privacy they provide to their users.

As opposed to previous works that tackle these privacy problems using a heuristic or a machine learning approach, in this thesis we follow a *statistical approach*. First, we characterize the system model and the behavior of the users of electronic services in terms of probabilities. Then, based on these models, we study how to improve the privacy of the users. A particular advantage of this approach is that our results (i.e., our improvements to privacy-preserving systems) have a consistency supported by theoretical guarantees. This means that, as long as the users of the system follow the theoretical behavioral models that we consider, we know that our designs are optimal (or highly effective). In many cases, heuristic or machine learning approaches cannot provide such guarantees.

We note that the privacy goals and implementations of mix-based anonymous communication systems and obfuscation-based location privacy mechanisms are very different. However, throughout the thesis we will see that the *mathematical models* of these two privacy-preserving technologies are surprisingly close.

Their main difference lies in which resources users sacrifice for privacy: mix-based anonymous communication systems generally trade in communication delay for privacy, while obfuscation-based location privacy mechanisms sacrifice quality of service.

We explain the two privacy problems we consider, and our contributions in each scenario, in the sections below.

1.1. Meta-Data Protection Against a Passive Eavesdropper

The first scenario that we study in this thesis is *meta-data leakage protection* against a passive eavesdropper. Consider the example that we mentioned above, where Alice is browsing a website with information about a rare disease. Assume that there is a passive eavesdropper, i.e., a party that observes the messages exchanged between Alice and the web server, but does not interfere in the communications (e.g., Alice's ISP). Even if Alice and the web server encrypt the messages they exchange, the eavesdropper can see meta-data such as Alice's and the web server IP addresses. From this information alone, the eavesdropper can infer that Alice is communicating with a web server that hosts a webpage with information about a rare disease, thus compromising Alice's privacy.

A solution to protect meta-data from observers is to use *anonymous communication systems* [1, 2]. These systems are built on top of the Internet protocols, and rely on combining re-routing and encryption techniques to hide communication meta-data. Re-routing consists in changing the normal path that a message would follow through the network, i.e., the set of nodes that relay the packet to its destination. This is necessary, since otherwise it would be easy for an adversary to track packets and identify who communicates with whom. On the other hand, encryption techniques can be used to change the appearance of messages at each relay, which prevents an adversary from tracking a message based on its appearance.

In terms of latency, we can broadly classify anonymous communication systems as high or low-latency. High-latency systems purposely delay the messages they relay to break timing patterns in the communications that could otherwise reveal that two users are communicating. This is the case of Chaum's mix [3] and other mix-based proposals [4–6]. These designs are appropriate for applications that do not require real-time communication, such as email. On the contrary, low-latency systems are designed for applications that only tolerate small delays, such as web browsing or instant messaging [7–11]. In terms of anonymity properties, allowing for higher delays is rewarding, since this delay allows to reduce timing correlations between the packets of the same connection, thus making it

harder for a passive eavesdropper to detect the communicating parties. As a consequence, high-latency anonymous communication systems defend against global passive adversaries (e.g., an ISP with global vision of the packets traversing the network). Low-latency schemes, on the contrary, can only provide strong privacy guarantees against local attackers [9] or have to resort to flooding the network with dummy traffic (fake traffic used to confuse the adversary, which consumes bandwidth) to defend against stronger adversaries [10]. In this thesis, we focus on the study of mix-based anonymous communication systems, a family of (generally high-latency) schemes that derive from Chaum's mix [3].

1.1.1. Mix-based Anonymous Communication Systems

Mix-based anonymous communication systems are a family of anonymization schemes that are built using special relaying nodes called mixes [3]. Mixes were proposed by Chaum in the seminal paper [3], and were refined in subsequent works [4,12–14]. Broadly speaking, a mix is a router that relays messages in such a way that it is not possible for an external observer to link incoming and outgoing packets. In order to achieve this, the mix performs two basics operations with the packets it relays: it delays them, and changes their appearance. Fig. 1.1 depicts a generic mix model, that works as follows: first, the mix gathers the messages it receives and stores them in its buffer or *pool* until a certain *flushing condition* is triggered. Then, the mix selects some messages from its pool according to a *pool selection strategy*, changes their appearance using cryptographic transformations, and outputs them in a random order to their corresponding recipients. The process that encompasses these operations (randomly delaying messages, changing their appearance, and forwarding them) is typically known as a *round* or batch of mixing. An external observer of the messages arriving and departing from the mix cannot trivially link them based on their appearance (due to the cryptographic transformations which prevent bit-wise linkability) or their arrival/departure times (due to the delay and batching of messages which prevent timing linkability). Thus, this observer can only *statistically* link the messages (e.g., know that the recipient of a particular message is among a set of recipients) and therefore the mix provides a certain degree of communication *anonymity*.

The mix itself knows the correspondences between the input and output messages (since it handles all the appearance and timing transformations). To avoid placing full trust in a single mix, implementations of mix-based anonymous communications systems build networks by connecting several of them. Messages are routed through the network, so that a single message traverses a number of mixes before reaching its destination. In [3], Chaum proposes to use a cascade of mixes (i.e., several mixes connected one after the other) chosen according to the network topology or trust. The sender encrypts the message using *layered encryption*, and each mix decrypts the outer layer, performs its operations, and

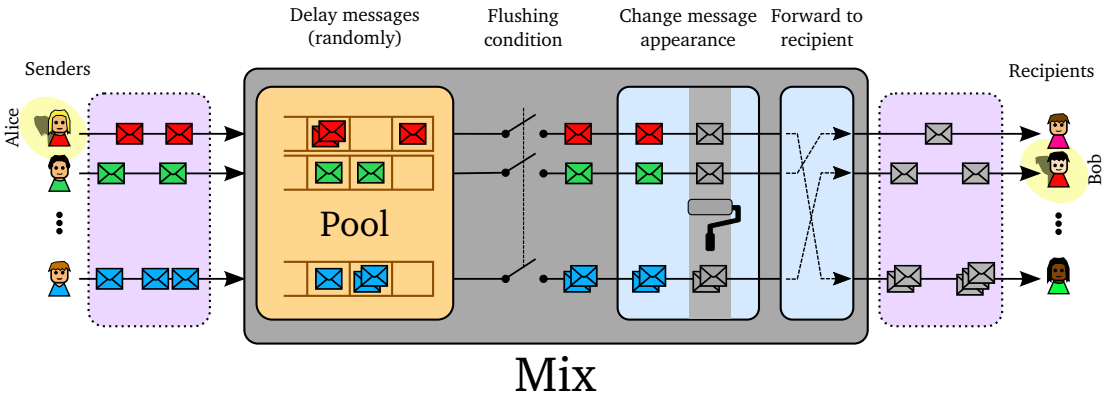


Figure 1.1: Mix model. The mix gathers messages from different senders and stores them in its pool. When a certain flushing condition is triggered, the mix selects some messages from its pool, transforms them cryptographically and forwards them to their recipients. An eavesdropper observing the traffic that traverses the mix cannot link the incoming and outgoing messages.

forwards messages to the next node. This way, the system provides anonymity against a global passive adversary as long as she is not in control of all of the mixes that the message traverses. Subsequent iterations of mixnets propose different network topologies (free-route networks, restricted routes) that provide different scalability, reliability and privacy properties [4, 6, 15–17].

Types of Mixes. Regarding their mode of operation, mixes can be broadly classified according to their flushing condition and their pool selection strategy [18]. The *flushing condition* is the event that causes messages to leave the mix. There are different types of mixes according to their flushing condition. *Threshold mixes* gather messages until a certain pre-determined number of them have been received. Then, they run their pool selection strategy and forward some of these messages to their destination. On the other hand, *timed mixes* output messages periodically according to a timer. Some designs, such as Mixmaster [4], implement more complex algorithms by combining these two flushing conditions. Others, known as *continuous mixes* [13], such as Kesdogan's Stop-and-Go mixes [12], get rid of the concept of "batches" and store each message independently for a certain amount of time that is sampled randomly from a delay distribution.

The *pool selection strategy* determines how the mix chooses messages from its pool when the flushing condition triggers. Even though all mixes have a "pool" where they store messages, the term *pool mixes* is normally reserved for mixes that have a non-zero probability of keeping some messages in its pool between rounds (i.e., those mixes that sometimes do not output all of their messages at the end of a round). Broadly speaking, we can distinguish between pool mixes that apply an independent delay to each message and those that do not. A typical example of the former are *binomial pool mixes* [14], that flip a biased coin for each message

at the end of each round to decide if it leaves the pool or not. Mixes that apply an independent delay to each message can be characterized by the probability density function of this delay, called *delay characteristic* function [13,19]. On the contrary, *deterministic mixes* [14] pick a fixed number of messages randomly from the pool and output them. This creates dependencies between the delays of the messages inside the pool (e.g., the fact that a message leaves the pool decreases the probability that another particular message has left in the same round).

Dummy Traffic. A common approach to improve the anonymity properties of mix-based systems is to include *dummy traffic* into the mix designs. Dummy messages are fake messages that are indistinguishable in appearance from real ones, and can either be generated by the users [20] or by the anonymity provider [18]. Since they look as real messages to the eyes of an external observer, dummy messages increase the adversary's uncertainty about who is the real sender/recipient of a message [21,22]. This increase in anonymity comes at an overhead cost, i.e., an increase in the bandwidth required to communicate.

Attacks on Mixes. When communications take place over a sufficiently long period of time, a malicious eavesdropper observing the flow of messages traversing the mixes can learn information about the communication preferences of the users by means of a *disclosure attack*. These communication preferences refer to who the communication partners of a particular sender are, or how frequently a sender sends messages to a particular recipient. The first Disclosure Attack [23,24] relies on graph theory to uncover the recipient set of a target user Alice. It identifies the set of Alice's contacts by seeking mutually disjoint sets of receivers among the recipient anonymity sets of the messages sent by Alice. The subfamily of Hitting Set Attacks [25,26] speeds up the search for Alice's messages recipients by restricting the search to unique minimal hitting sets.

The Statistical Disclosure Attack (SDA), originally proposed by Danezis [27], and its sequels [22,28,29], estimate Alice's sending profile by averaging the probability distributions describing the recipient anonymity set [30] of her messages. Mathewson and Dingledine improved Danezis' SDA by extending it to a more general scenario and to more complex mixing algorithms [22].

Troncoso et al. proposed in [31] two attacks: the Perfect Matching Disclosure Attack (PMDA) and the Normalized Statistical Disclosure Attack (NSDA). These attacks can only be used when the mixes flush all the messages in their pool at the end of a round. The attacks exploit that, in this case, the relationship between sent and received messages in a round must be one-to-one. PMDA accounts for this interdependency by searching for perfect matchings in the underlying bipartite graph representing a mix round, while NSDA normalizes the adjacency matrix representing this graph.

Danezis and Troncoso propose Vida attack [32], where they use Bayesian sampling techniques to co-infer users' profiles and de-anonymize messages. The

Bayesian approach outputs samples from the distribution of all possible sending profiles, which in turn allows to infer reliable error estimates. However, Vida requires the adversary to repeatedly seek for perfect matchings, increasing the computational requirements of the attack. Finally, Pérez-González and Troncoso propose the Least-Squares Disclosure Attack (LSDA) [33], that estimates the users' communication profiles as the solution of a least-squares problem.

1.1.2. Our Contributions

Our main contributions in this topic are separated into two chapters of the thesis:

- **Chapter 2: Limits of Dummy Traffic Protection in Anonymous Communication Systems.** We analyze the effect of dummy traffic in the anonymity granted by pool mixes. We provide closed-form expressions for the privacy of the users as a function of the system parameters. This allows us to design *optimal dummy strategies*, i.e., find the best way of using a restricted budget of dummy traffic towards achieving a specific privacy goal. We demonstrate the feasibility of our approach on two privacy objectives: increase the protection of all the users by a constant factor, and maximize the minimum protection of all the users in the system.

  This chapter is adapted by permission from Springer Nature Customer Service Centre GmbH: Simon Oya, Carmela Troncoso, and Fernando Pérez-González. Do dummies pay off? limits of dummy traffic protection in anonymous communications. In Privacy Enhancing Technologies*, volume 8555 of* Lecture Notes in Computer Science*, pages 204–223.* Springer International Publishing*, 2014.*

- **Chapter 3: Design of Pool Mixes Against Profiling Attacks in Real Conditions.** We study the performance of pool mixes that independently delay each message under realistic user behavior. We propose a statistical model for user behavior that captures complex behavioral traits. Then, we use this model to analyze the anonymity of pool mixes and study the *delay characteristic* that maximizes this anonymity. We evaluate our behavioral model and our delay characteristic designs using real traces, showing that they outperform previous proposals.

  This chapter is adapted with permission from IEEE: Simon Oya, Fernando Pérez-González, and Carmela Troncoso. Design of pool mixes against profiling attacks in real conditions. IEEE/ACM Transactions on Networking*, 24(6):3662–3675, 2016.*

We have made other contributions on this topic, that for brevity we do not present in this thesis:

- **Meet the Family of Statistical Disclosure Attacks [34].** We compare different variants of the Statistical Disclosure Attack (SDA) found in the literature [22, 27], propose two new attacks and show their relation with the Least Squares Disclosure Attack (LSDA) [33]. We prove analytically that LSDA asymptotically outperforms the most sophisticated variant of the SDA, and evaluate this finding empirically.

- **A Least Squares Approach to the Static Traffic Analysis of High-Latency Anonymous Communication Systems [35].** We formalize the derivation of the Least Squares Disclosure Attack (LSDA) in pool mixes without the assumptions of previous works [33, 36], and propose an algorithm to compute a constrained version of LSDA (C-LSDA) that improves its performance. We extend the theoretical analysis of LSDA found in [33,36], and empirically evaluate the accuracy of this theoretical formula.

- **Understanding the Effects of Real-World Behavior in Statistical Disclosure Attacks [37].** We update the analysis of LSDA in threshold and timed mixes (without delay between rounds) under realistic user behavior. We relax some unrealistic assumptions of previous works regarding how users act, and obtain a new closed-form expression of the performance of LSDA. We evaluate this theoretical estimation of LSDA's performance using three real-world datasets, confirming that it accurately predicts the actual performance of LSDA in real cases.

- **Filter Design for Delay-Based Anonymous Communications [38].** We extend the ideas in our previous work [19] regarding the design of the delay characteristic of pool mixes in real scenarios, and show that this problem is connected to filter design problems. We show that the optimal delay characteristic depends on the adversary observation time, and derive solutions that are optimal either against long-term or against short-term attacks. Finally, we use the connections between pool mix design and filter design to extract conclusions about the overall performance of cascades of mixes and propose a decentralized implementation of the binomial pool mix.

1.2. Privacy Against an Adversarial Service Provider

In the second part of this thesis, we study the problem of communications with an adversarial service provider. Consider a user that is interested in getting a service from an honest-but-curious provider. For this, she sends a query to the provider, and the provider replies with the requested information. In many cases, the user query contains sensitive information (e.g., the user browsing preferences, or her actual GPS location). However, the user cannot encrypt the query to hide

it against the provider, otherwise the provider would not be able to generate any useful response. Thus, the user has to rely on other techniques to hide her sensitive information and at the same time get a useful response from the service provider.

In this thesis, we focus on one instance of this privacy problem: the case of providing privacy to the users of Location Based Services (LBS). These services rely on the user's real location to operate (e.g., they search for nearby points-of-interest, register location check-ins, or count calories based on the user's movements). The real user location is many times sensitive, and thus sharing it with the service provider violates the user's privacy. We illustrate this privacy problem with an example. Imagine that Alice is at a hospital and wants to find out where the closest bar is. For this, she sends a query to a Location Based Service (LBS) using her smartphone. The LBS replies to Alice, but also learns that she is at a hospital. This is a violation of Alice's privacy, since it might suggest that Alice has a particular disease.

Fortunately, the privacy community has developed many solutions that partially hide the location information from service providers. This is known as providing *location privacy* [39, 40]. Duckham and Kulik [41] broadly classify location privacy protection strategies into four types: *regulatory strategies* (e.g., government rules on how to use personal data), *privacy policies* (particular agreements between the individuals that provide their location data and the entities that receive it), *anonymity* (e.g., sharing the location data of many individuals while replacing the identity of each of them with a pseudonym), and *obfuscation* (share the location data with reduced quality). In addition to these approaches, other works propose *cryptographic* solutions for location privacy [42–44]. In this thesis, we focus exclusively on obfuscation-based location privacy, and use the term Location Privacy Preserving Mechanism (LPPM) to refer to the tool that individuals use to obfuscate their location data.

1.2.1. Obfuscation-Based Location Privacy

There are many strategies that deliberately degrade the user's location data, while preserving some utility of the query results. One of the most common techniques is *perturbation*. Perturbation-based LPPMs generate obfuscated locations by adding random noise to the real locations of the user. This noise can be sampled from a known distribution (e.g., 2-dimensional Laplacian noise [45]) or characterized by a probability density function (pdf) built ad-hoc for the situation (e.g., the pdfs that result from solving an optimization problem [46]). Figure 1.2 illustrates a scenario where Alice uses a perturbation mechanism. Here, Alice wants to know the bar that is closest to her location. However, she does not want to reveal her real location to the LBS. Therefore, she relies on a perturbation-based LPPM to generate an alternative location, which is achieved by adding

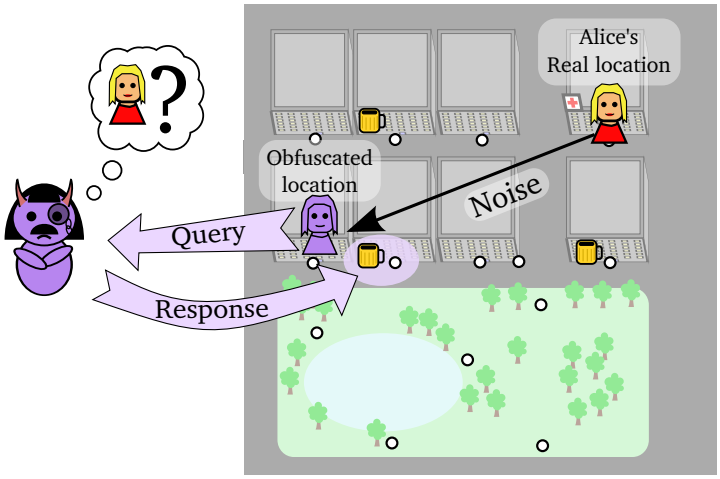


Figure 1.2: Example of a perturbation-based LPPM. Alice wants to know the bar that is closest to herself, but does not want to reveal her real location. In order to do so, she uses a Location Privacy Preserving Mechanism (LPPM) to generate an obfuscated location and performs the query with respect to this location. This impedes the service provider from learning Alice's whereabouts. Thus, she gets privacy, but she also loses utility, since the bar that is closest to her obfuscated location is not the one that is closest to her real location.

noise to her real location. Alice performs the query with her obfuscated location. The LBS observes the obfuscated location and knows that Alice has generated this fake location by adding noise to her real one. Even though the LBS knows that Alice's real location lies close to the reported location, it is not possible for it to pinpoint Alice's real location in the map. Thus, Alice achieves a certain degree of privacy against the LBS. However, she also loses some quality of service, since the bars that are closest to her actual location might not be the ones that are closest to the perturbed location.

Another type of obfuscation-based LPPMs are *cloaking* mechanisms. These LPPMs build a "cloaking region" that contains several other users and/or location venues, and then perform the query with respect to this area. This way, the adversary receiving the queries from several users inside the same region cannot identify the source of each query. One complication with cloaking mechanisms is that they require cooperation among users [47] or placing trust in a centralized anonymizer server to build the cloaking region [48–50]. Other type of obfuscation techniques are *hiding* mechanisms, which decide randomly whether to perform the query using the real location, or to not perform it at all [51]. Finally, *dummy-based* techniques generate multiple fake locations alongside the real one, and perform the query for each of them [52–54]. The adversary is not able to know for sure which of the queries is the real one, and the user gets privacy in exchange for communication overhead.

Classifications of LPPMs. LPPMs can be classified according to their architecture as *centralized* or *decentralized* LPPMs. Centralized LPPMs require a trusted third party that gathers location information from many users and processes it before releasing it to the service provider (e.g., the aforementioned cloaking mechanisms [48–50]). Decentralized LPPMs are *user-centric*, i.e., users run these LPPMs individually on their mobile devices to generate the obfuscated locations that they send to the service provider. In this thesis, we focus only on user-centric LPPMs.

On the other hand, LPPMs can be implemented in an *online* or an *offline* manner. Online LPPMs generate obfuscated locations on the go as soon as the user requires querying the LBS. These LPPMs are useful for applications such as location-based queries or proximity services for social networks. Offline LPPMs, on the other hand, receive the complete location trace of users and obfuscate it as a whole (e.g., sharing a database of past location traces with an analyst). In this thesis, we consider online LPPMs.

Measuring Location Privacy. Location privacy protection mechanisms are designed with a privacy goal in mind that is expressed in terms of location privacy metrics. Thus, defining how to quantify location privacy is crucial towards the development of defense strategies.

One of the most popular location privacy metric of early works is k-anonymity. This notion is borrowed from the database privacy field [55] and is based on creating a cloaking region for k users, such that an adversary receiving one of the queries cannot attribute to which individual (out of the k users) it belongs to. This notion was first adopted by [56] and it was used in many follow-ups [48, 57–61]. However, it became clear later that k-anonymity provides *query anonymity* but not *location privacy*, as reported in [62]. This is easy to see: if the k users are all together in a small region, even though they have anonymity against the LBS, their location is trivially revealed.

Taking inspiration from previous works [39, 40, 63], in 2011 Shokri et al. [51] propose a framework to quantitatively evaluate LPPMs, claiming that location privacy should be measured as the adversary's *correctness*. This metric can be defined as the average distance between the adversary's estimation of the user's real location, and the user's real location. Here, the concept of "distance" can be tailored to each particular application, e.g., it can be the Euclidean distance, but also a semantic distance that takes into account the sensitivity of certain locations [64]. This notion of privacy was widely adopted in most of the works that followed [45,46,65–67] and became a standard location privacy metric. One of the disadvantages of the adversary *correctness* is that it depends on the particular adversary that is taken into account and her prior knowledge about the user's whereabouts.

In 2013, Andrés et al. propose *geo-indistinguishability* [45], an extension of

the widely known notion of differential privacy [68, 69] to 2-dimensional spaces. Geo-indistinguishability is based on the idea that two users that are close should be indistinguishable if they generate obfuscated locations with a similar spatial distribution. This notion has been broadly adopted in the location privacy community [65–67, 70–75], due to its appealing features. One of the most relevant properties of geo-indistinguishability is that it is adversary-agnostic, i.e., it is a privacy property guaranteed by the LPPM only, regardless of the amount of information that the adversary owns about the user's whereabouts (contrary to Shokri's correctness).

1.2.2. Our Contributions

Our location privacy contributions appear in two chapters:

- **Chapter 4: Revisiting Location Privacy Metrics.** We study the traditional LPPM evaluation approach [51], where privacy and utility are measured as the adversary correctness and the average quality loss, respectively. We show that there are infinite optimal mechanisms according to these metrics, and find one mechanism that, while being optimal, is unsuitable for the user in terms of usability and privacy. This demonstrates that judging privacy as correctness alone can be dangerous from a privacy standpoint. We claim that, in order to properly assess the privacy properties of an LPPM, we must use complementary privacy metrics, and particularly advocate the advantages of using the conditional entropy in this regard. We propose efficient techniques to optimize mechanisms with respect to the conditional entropy, developing an LPPM that we call ExPost. We evaluate the performance of ExPost and other LPPMs in terms of different privacy and quality loss metrics, showing that no mechanism is optimal in terms of all privacy metrics, and that we cannot rely solely on the correctness metric to assess the performance of an LPPM.

  This chapter is adapted with permission from ACM: Simon Oya, Carmela Troncoso, and Fernando Pérez-González. Back to the drawing board: Revisiting the design of optimal location privacy-preserving mechanisms. In Proc. of Computer and Communications Security *(CCS), pages 1959–1972. ACM, 2017.*

- **Chapter 5: Rethinking Location Privacy for Unknown Mobility Behaviors.** Previous location privacy works largely consider that the statistics that characterize user mobility are fixed and known a-priori. Thus, they hardwire these characteristics on the mobility models that they use to design LPPMs. In this chapter, we challenge this hardwired approach, and show that hardwired LPPMs perform much worse when evaluated in data that deviates statistically from such models (which is expected to occur in practice). To solve this issue, we propose blank-slate models for user

mobility. These models are not completely determined by training information, but are updated a-posteriori as users query the LBS. We leverage this blank-slate model to build a new type of LPPMs that we call Profile Estimation-Based (PEB)-LPPMs. We show using real datasets that PEB-LPPMs outperform previous hardwired proposals when the training data does not capture individual users' mobility traits.

This chapter is adapted with permission from IEEE: Simon Oya, Carmela Troncoso, and Fernando Pérez-González. Rethinking location privacy for unknown mobility behaviors. In IEEE European Symposium on Security and Privacy (EuroS&P)*, IEEE 2019.*

We have made an additional contribution to the topic of location privacy, that we have left out of the thesis for space reasons.

- **Is Geo-Indistinguishability What You Are Looking for? [76].** We study the geo-indistinguishability privacy notion and identify that most of the previous works blindly rely on geo-indistinguishability LPPMs to provide location privacy, without actually quantifying the amount of privacy that these LPPMs provide to the users. We propose an alternative interpretation of geo-indistinguishability as a lower bound on the adversary's probability of error, and use this more intuitive notion to show that the amount of noise that is required to provide an acceptable privacy level using geo-indistinguishability is, for most applications, prohibitive. This challenges the usage of geo-indistinguishability as a "dogma" and urges to find settings where this notion can be implemented at a reasonable cost, such as relying on centralized LPPMs.

Part I

Mix-Based Anonymous Communication Systems

Chapter 2

Limits of Dummy Traffic Protection in Anonymous Communication Systems

2.1. Introduction

In the previous chapter we have seen that mixes, anonymous relays that hide the correspondence between the messages they receive and the messages they output using cryptographic techniques and delay [3], are susceptible to a plethora of disclosure attacks [22–24, 27–33]. These attacks allow a passive adversary to infer the long-term communication preferences of the users (i.e., their communication *profiles*) by observing the traffic flows that arrive to and depart from the mix.

We have also seen that a common approach to improve users' protection against profiling is to introduce dummy traffic. The effectiveness of this countermeasure has been studied theoretically from the perspective of individual messages in [77]. With respect to profiling, dummy traffic has been tackled in [21, 22], where the authors empirically compute the number of rounds that the attacker takes to correctly identify some or all the recipients of a sender. The analyses in [21, 22] are limited in two aspects. On the one hand, the results strongly depend on the specific cases considered in the experiments, and it is difficult to get insight on their applicability to other scenarios. On the other hand, the analyses only consider the ability of the adversary in identifying communication partners,

This chapter is adapted by permission from Springer Nature Customer Service Centre GmbH: Simon Oya, Carmela Troncoso, and Fernando Pérez-González. Do dummies pay off? limits of dummy traffic protection in anonymous communications. In Privacy Enhancing Technologies*, volume 8555 of* Lecture Notes in Computer Science*, pages 204–223.* Springer International Publishing*, 2014.*

but not her accuracy at estimating the intensity of the communication; i.e., the users' profiles.

In this chapter of the thesis, we study the effectiveness of dummy traffic against profiling attacks in mix-based anonymous communication systems. Our goal is to obtain analytical (rather than only empirical) results, so that they are generalizable to a wide range of high-latency anonymous communication schemes, and provide the analyst with a bound on the protection achievable through dummy traffic. Our analysis is based on the least squares approach introduced in [33].

Another shortcoming of previous works [21, 22, 77] is that the proposed evaluation strategies cannot be used to guide the design of effective dummy generation strategies, which is recognized to be a hard problem [5]. This has lead the deployed high latency anonymous communication systems to either implement arbitrary dummy strategies [4] or no dummy traffic at all [5]. Our methodology can be used to support the design of dummy strategies by approaching strategy selection as an optimization problem in which the error of the adversary is maximized. The optimization criteria can be chosen by the designer to satisfy different privacy objectives, e.g., balancing the protection among users, or favoring individual users or relationships.

We illustrate the operation of our methodology using a timed binomial pool mix. We provide a performance analysis of this mixing strategy in presence of both sender-based and mix-based dummy traffic, showing that their contribution to the adversary's error can be decoupled and analyzed independently. Departing from this analysis, we design dummy traffic strategies according to two privacy criteria: increasing the estimation error for all the relationships by a constant factor, and guaranteeing a minimum estimation error for any relationship. By hiding relationships, both criteria hinder adversary's effort to infer user profiles.

This chapter is organized as follows. Section 2.2 introduces the system and adversary model that we assume. In Section 2.3 we derive a least-squares estimator of the users' sending profiles in dummy-based anonymization systems. We analyze the performance of this estimator in Section 2.4 when the anonymous channel is a timed binomial pool mix. The result of this analysis is used in Sect. 2.5 to design optimal dummy strategies, and evaluated in Sect. 2.6. We discuss practical aspects of our method in Sect. 2.7 and finally conclude in Sect. 2.8.

2.2. System Model and Notation

In this section, we introduce the system and adversary model that we assume in this chapter, together with our privacy metric and the notation we use (summarized in Table 2.1).

System Model. Our system consists of N senders, designated by index $i \in \{1, 2, \ldots, N\}$, that communicate with M receivers, designated by index $j \in \{1, 2, \ldots, M\}$, through a mix-based anonymous communication system implementing a pool. Messages in the system may be real or dummy messages: decoy messages indistinguishable from real traffic. We consider two types of *dummy traffic*:

- **Sender-based dummies:** senders may send *dummy* messages to the mix along with their *real* messages. Sender-based dummies can be recognized and discarded by the mix.
- **Mix-based dummies:** the mix-based system may send *dummy* messages to the receivers along with the *real* messages from the senders. Receivers are able to identify dummy messages and discard them.

The system operates in batches that we call *communication rounds*. The operation of the mix, i.e., its *batching strategy*, can be characterized by the model in Figure 2.1, that encompasses the following steps:

1. The senders forward both their real and dummy messages to the mix. We use x_i^r to denote the total number of messages generated by user i in round r. The real and dummy messages are denoted by $x_{\lambda,i}^r$ and $x_{\delta,i}^r$, respectively. Note that $x_i^r = x_{\lambda,i}^r + x_{\delta,i}^r$.
2. The mix gathers these messages from the senders, identifies the dummy messages, and discards them (Stage 1).
3. The mix assigns to each of the real messages a waiting time (in rounds) chosen according to a *delay characteristic* d, and stores them in its pool. When a certain *flushing condition* triggers (e.g., a timer expires), the mix selects from the pool the messages whose waiting time has expired and forwards them to the next step. The mix decreases in one unit the waiting time of the messages that remain in the pool. These messages will be mixed with the ones arriving in subsequent rounds (Stage 2).
4. Messages leaving the pool traverse a mixing block, which changes their appearance cryptographically to avoid bit-wise linkability. The messages are re-organized according to their corresponding recipient (Stage 3). We use $y_{j,i}^r$ to denote the number of real messages from sender i that are addressed to receiver j that leave the pool in round r. We group these messages by sender as $z_i^r \doteq \sum_{j=1}^{M} y_{j,i}^r$ and by recipient as $y_{\lambda,j} = \sum_{i=1}^{N} y_{j,i}^r$.
5. The mix combines the real messages (from the previous step) with mix-based dummies before forwarding them to their recipients (Stage 4). We use $y_{\delta,j}^r$ to denote the number of dummies sent by the mix to recipient

Table 2.1: Summary of notation.

Symbol	Meaning
N	Number of senders, denoted by $i \in \{1, \cdots, N\}$.
M	Number of receivers, denoted by $j \in \{1, \cdots, M\}$.
ρ	Number of rounds observed by the adversary, $r \in \{1, \cdots, \rho\}$.
$p_{j,i}$	Probability that a (real) message from sender i is addressed to receiver j.
$\hat{p}_{j,i}$	Adversary's estimation of $p_{j,i}$.
$x^r_{\lambda,i}$ ($x^r_{\delta,i}$)	Number of real (dummy) messages sent by user i in round r.
x^r_i	Total number of messages sent by sender i in round r. $x^r_i = x^r_{\lambda,i} + x^r_{\delta,i}$.
z^r_i	Number of real messages sent by i that leave the pool in round r.
$y^r_{j,i}$	Number of real messages from i to j that leave the pool in round r.
$y^r_{\lambda,j}$ ($y^r_{\delta,j}$)	Number of real (dummy) messages received by j in round r.
y^r_j	Total number of messages received by j in round r. $y^r_j = y^r_{\lambda,j} + y^r_{\delta,j}$.
d_k	Probability that a message is delayed k rounds in the pool.
P_{λ_i}	Probability that a message sent by user i is real.
$\delta^{\texttt{MIX}}_j$	Average number of mix-dummies received by j each round, $\mathrm{E}\left\{Y^r_{\delta,j}\right\} = \delta^{\texttt{MIX}}_j$.
$\mathrm{MSE}_{j,i}$	Privacy metric: adversary's estimation error of $p_{j,i}$, as defined in (2.1).
$\mathbf{q}_i$	Sending profile of user i, $\mathbf{q}_i \doteq [p_{1,i}, p_{2,i}, \cdots, p_{M,i}]^T$.
$\mathbf{p}_j$	Vector of probabilities per receiver, $\mathbf{p}_j \doteq [p_{j,1}, \cdots, p_{j,N}]^T$.
$\mathbf{P}$	Matrix of all probabilities, $\mathbf{P} \doteq [\mathbf{p}_1, \cdots, \mathbf{p}_M]$.
$\mathbf{X}$	Matrix with all the input messages, $(\mathbf{X})_{r,i} \doteq x^r_i$.
$\mathbf{Z}$	Matrix with all the messages that leave the pool, $(\mathbf{Z})_{r,i} \doteq z^r_i$.
$\hat{\mathbf{Z}}$	Adversary estimation of $\mathbf{Z}$, as defined in (2.6).
$\mathbf{y}_j$	Column vector of all the messages received by j, $\mathbf{y}_j \doteq [y^1_j, y^2_j, \cdots, y^\rho_j]^T$.
$\mathbf{Y}$	Matrix with all the output messages, $(\mathbf{Y})_{r,j} \doteq y^r_j$.
$\mathbf{Y}_\delta$	Matrix with all the mix-based dummies, $(\mathbf{Y}_\delta)_{r,j} \doteq y^r_{\delta,j}$.
$\mathbf{D}$	Convolution matrix of the delay characteristic, shown in (2.4).
$\mathbf{P}_\lambda$	Diagonal matrix with all the probabilities of real message, $(\mathbf{P}_\lambda)_{i,i} \doteq P_{\lambda_i}$.
$\boldsymbol{\delta}^{\texttt{MIX}}$	Vector of all the avg- mix-dummies per receiver, $\boldsymbol{\delta}^{\texttt{MIX}} \doteq [\delta^{\texttt{MIX}}_1, \delta^{\texttt{MIX}}_2, \cdots, \delta^{\texttt{MIX}}_M]^T$.

j in round r. The total number of messages received by j in round r is $y^r_j \doteq y^r_{\lambda,j} + y^r_{\delta,j}$.

6. The recipients decrypt the messages they receive, discard the mix-based dummies, and keep the real ones.

We also define the following vectors and matrices, which shall come handy later: matrix $\mathbf{X}$ is a $\rho \times N$ matrix which contains all the input observations, i.e., its (r,i)-th element is x^r_i. Similarly, matrix $\mathbf{Z}$ contains in its (r,i)-th position the number of messages from sender i that leave in round r, z^r_i. Matrix $\mathbf{Y}$ is a $\rho \times M$ matrix that contains all the output observations, i.e., its (r,j)-th element is y^r_j. Matrix $\mathbf{Y}_\delta$ contains only the mix-based dummy messages that leave the mix, i.e., its (r,j)-th element is $y^r_{\delta,j}$. Finally, $\mathbf{y}_j$ is a $\rho \times 1$ column vector with the number of messages that leave for receiver j in each round, i.e., $\mathbf{y}_j \doteq [y^1_j, y^2_j, \cdots, y^\rho_j]^T$.

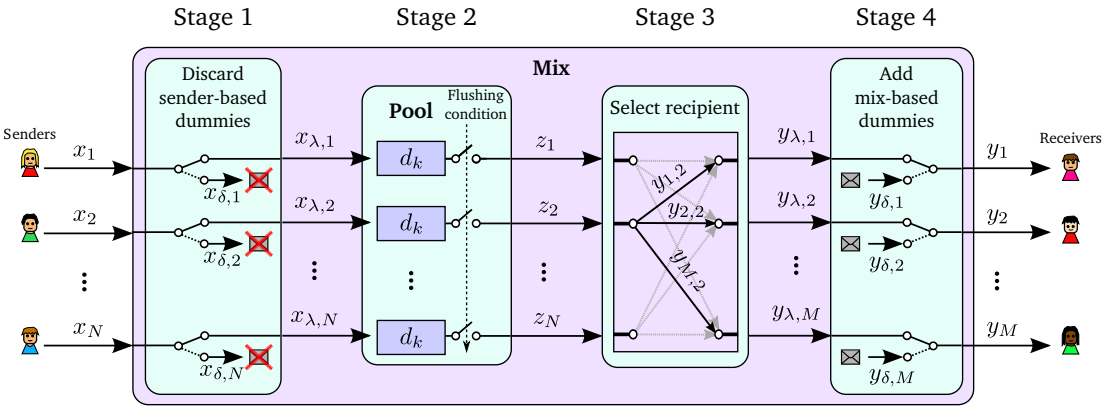


Figure 2.1: Abstract model of a round in a mix-based anonymous communication channel (we omit the subscript r for the sake of clarity). A global passive adversary is only able to see the messages arriving and leaving the mix (i.e., $x_1^r, x_2^r, \cdots, x_N^r$ and $y_1^r, y_2^r, \cdots, y_M^r$) but is not aware of what happens inside of it.

We model the sending behavior of users in our population with two parameters:

- **Probability of real message:** the probability of real messages models how frequently users send real messages, and is denoted by $P_{\lambda_i}, i = 1, \cdots, N$. In other words, we assume that each message sent by i is real with probability P_{λ_i}, and dummy otherwise, independently of the rest of the messages. We make no assumptions on the values of P_{λ_i} other than $0 \leq P_{\lambda_i} \leq 1$, and that the probabilities of real messages are stationary during the observation period.

- **Sender profile:** the sender profile of user i models this sender's choice of recipients for her messages. It is defined as the vector $\mathbf{q}_i \doteq [p_{1,i}, p_{2,i}, \cdots, p_{M,i}]^T$, where $p_{j,i}$ denotes the probability that sender i sends a real message to receiver j. We also define the unnormalized receiver profile $\mathbf{p}_j \doteq [p_{j,1}, \cdots, p_{j,N}]^T$ and the matrix containing all transition probabilities $\mathbf{P} \doteq [\mathbf{p}_1, \cdots, \mathbf{p}_M]$. We make no assumptions on the shape of the sender profiles other than $\mathbf{q}_i$ is in $\mathcal{P}$, the probability simplex in $\mathbb{R}^M$, i.e., $\mathcal{P} \doteq \left\{\mathbf{r} \in \mathbb{R}^M : r_i \geq 0, \sum_{i=1}^M r_i = 1\right\}$. We use $\mathbf{P} \in \mathcal{P}^N$ to denote that each sender profile belongs to $\mathcal{P}$. We assume, nevertheless, that users' behavior is stationary during the observation period (the transition probabilities $p_{j,i}$ do not change between rounds), independent (the behavior of a user does not affect the behavior of the others) and memoryless (the messages sent by a user in a round do not affect the behavior of that user in subsequent rounds). We discuss the implications of the hypotheses above being false in Sect. 2.7.

The behavior of the mix-based anonymous communication channel is modeled by four parameters:

- **Flushing condition:** the flushing condition is an event, e.g., the arrival of a message (theshold mix) or the expiration of a timeout (timed mix), that causes the mix to forward some of the messages it has stored in its pool to their recipients.
- **Delay characteristic:** the delay characteristic models how messages are chosen to leave the pool. The delay characteristic is defined by the *probability mass function* of the delay, measured in rounds. The probability that a message is delayed k rounds inside the pool is denoted by d_k ($k \geq 0$). We assume that the delay characteristic is *stationary*, i.e., the probability that a message that arrives to the pool in round r leaves in round $s \geq r$ only depends on the difference $s - r$. We discuss the implications of this assumption in Sect. 2.7. We do not assume any particular shape for this distribution, besides $d_k \geq 0$ and $\sum_{k=0}^{\infty} d_k = 1$.
- **Mix dummy characteristics:** as explained above, the distributions that model each of the random variables $Y_{\delta,j}^r$ ($\forall j, r$) characterize the amount of mix-based dummies sent to receivers. We assume that $Y_{\delta,j}^r$ is stationary, i.e., its expectation, denoted $\delta_j^{\texttt{MIX}} \doteq \mathrm{E}\left\{Y_{\delta,j}^r\right\}$, does not change with time.

Adversary Model and Privacy Metrics. We consider a global passive adversary that observes the system during ρ rounds. The adversary is able to see the identity of each sender and receiver communicating through the mix, but she is not able to link any two messages by their content nor distinguish between real and dummy messages. We assume that the adversary knows all the parameters of the system (e.g., the delay characteristic d_k, the parameters modeling the generation of dummy messages P_{λ_i} and the distributions $Y_{\delta,j}^r$). The goal of the adversary is to infer the sending profiles of the users in the system from the observations, i.e., to obtain an estimator $\hat{p}_{j,i}$ of the probabilities $p_{j,i}$ given the input and output observations x_i^r and y_j^r, for every $i \in \{1, 2, \cdots, N\}$, $j \in \{1, 2, \cdots, M\}$ and $r \in \{1, 2, \cdots, \rho\}$.

We use the Mean Squared Error (MSE) of the adversary estimation as our privacy metric. We define this error, for each bilateral relationship between sender i and receiver j, as

$$\mathrm{MSE}_{j,i} \doteq \mathrm{E}\left\{|\hat{p}_{j,i} - p_{j,i}|^2\right\} . \tag{2.1}$$

The suitability of the estimation error as a privacy metric is thoroughly discussed in [78], but the intuition is simple: a larger estimation error means that the relationship between i and j is more protected against the adversary.

The metric in (2.1) depends on the particular estimator $\hat{p}_{j,i}$ that the adversary computes, and thus it is important to choose this estimator appropriately. The

long-term disclosure attacks proposed in the literature that are applicable to the general scenario we have presented are the attacks belonging to the so-called Statistical Disclosure Attack (SDA) family [22, 34, 35, 79], the Perfect Matching Disclosure Attack (PMDA) [31] and the Bayesian inference attack (Vida) [32]. We do not consider other attacks such as the Disclosure Attack [23] or the Hitting Set Disclosure Attack [25], since they estimate the exact set of contacts of each sender instead of the intensity of the communications of such sender with each of her contacts. We also leave the Two-Sided SDA [28] out of our study, since it is only applicable under some assumptions on how users reply to messages.

The SDA-based attacks obtain the estimator $\hat{p}_{j,i}$ by solving a *linear problem* that is built using the observations. Due to their mathematical simplicity, they can easily be extended to pool mixes with dummies [21,22]. PMDA and Vida work by finding *matchings* in the system, i.e., studying the possible correspondences between all messages entering and leaving the mix. PMDA is based on looking for the most probable matching, while Vida iterates by sampling matchings given the observations. In this sense, these two attacks follow a message-based approach, which they then use to estimate the sending profiles. In principle, we could think of extending PMDA and Vida to work in pool mixes with dummy traffic. However, finding matchings in a pool mix requires processing the whole trace *at once*, since the pool introduces dependencies between rounds. This renders PMDA and Vida computationally prohibitive against pool mixes. We therefore limit our choice to the attacks of the SDA family. From this family, the Least Squares Disclosure Attack (LSDA) has been proven to outperform all its relatives [34]. Therefore, we use the performance of LSDA as our metric for anonymity. We note however that, even though it outperforms any known feasible attack, LSDA is not necessarily the optimal attack against pool mixes and better non-linear attacks may appear in the future. Nevertheless, this is the first work to study optimal dummy strategies in pool mixes against profiling attacks and, hence, our results shall serve as baseline for future proposals.

2.3. A Least-Squares Profile Estimator for Dummy-based Systems

LSDA has been originally proposed for threshold mixes without delay between rounds [33], and extended to pool mixes in [36]. We now apply the methodology of [80] to derive a least squares estimator of the probabilities $p_{j,i}$ in a pool mix with dummy traffic. We do this by looking for the matrix of probabilities $\mathbf{P}$ that minimizes the Mean Squared Error (MSE) between the random matrix $\mathbf{Y}$ and its expected value given the inputs $\mathbf{X}$. We define the LSDA estimator of $\mathbf{P}$ as

$$\hat{\mathbf{P}} = \underset{\mathbf{P}\in\mathcal{P}^N}{\operatorname{argmin}} \left|\left|\mathbf{Y} - \mathrm{E}\left\{\mathbf{Y}|\mathbf{X}\right\}\right|\right|^2 . \tag{2.2}$$

In the expression above, the minimization is done among the probabilities $\mathbf{P}$, which are part of the expected value $\mathrm{E}\{\mathbf{Y}|\mathbf{X}\}$. In order to solve this minimization problem, we need to find an expression of $\mathrm{E}\{\mathbf{Y}|\mathbf{X}\}$ in terms of $\mathbf{P}$. We first consider the (r,j)-th element of this matrix, i.e., $\mathrm{E}\left\{Y_j^r|\mathbf{X}\right\}$. Note that the total number of messages received by j in round r is a combination of real and dummy messages, and the real messages are a combination of messages from all senders. Mathematically, we can write $Y_j^r = \sum_{i=1}^{N} Y_{j,i}^r + Y_{\delta,j}^r$ (see Fig. 2.1).

In Appendix 2.A we show that $\mathrm{E}\left\{Y_{j,i}^r|\mathbf{X}\right\} = \sum_{s=1}^{r} x_i^s \cdot d_{r-s} \cdot P_{\lambda_i} \cdot p_{j,i}$. Using this and the fact that $\mathrm{E}\left\{Y_{\delta,j}^r|\mathbf{X}\right\} = \mathrm{E}\left\{Y_{\delta,j}^r\right\} \doteq \delta_j^{\texttt{MIX}}$, we can write

$$\mathrm{E}\left\{Y_j^r|\mathbf{X}\right\} = \sum_{i=1}^{N}\sum_{s=1}^{r} x_i^s \cdot d_{r-s} \cdot P_{\lambda_i} \cdot p_{j,i} + \delta_j^{\texttt{MIX}}\,. \tag{2.3}$$

We can express this result in matricial form, i.e., find a closed-form expression for $\mathrm{E}\{\mathbf{Y}|\mathbf{X}\}$. If we define the convolution matrix

$$\mathbf{D} \doteq \begin{bmatrix} d_0 & 0 & 0 & \cdots & 0 \\ d_1 & d_0 & 0 & \cdots & 0 \\ d_2 & d_1 & d_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{\rho-1} & d_{\rho-2} & d_{\rho-3} & \cdots & d_0 \end{bmatrix}, \tag{2.4}$$

then $\mathrm{E}\{\mathbf{Y}|\mathbf{X}\}$ can be written as

$$\mathrm{E}\{\mathbf{Y}|\mathbf{X}\} = \mathbf{D}\cdot\mathbf{X}\cdot\mathbf{P}_\lambda\cdot\mathbf{P} + \mathbf{1}_\rho\cdot(\boldsymbol{\delta}^{\texttt{MIX}})^T\,, \tag{2.5}$$

where we have used that $\mathrm{E}\{\mathbf{Y}_\delta\} = \mathbf{1}_\rho\cdot(\boldsymbol{\delta}^{\texttt{MIX}})^T$ where $\boldsymbol{\delta}^{\texttt{MIX}} \doteq [\delta_1^{\texttt{MIX}}, \delta_2^{\texttt{MIX}}, \cdots, \delta_M^{\texttt{MIX}}]^T$.

Note that we can define the adversary's estimation of $\mathbf{Z}$ as $\hat{\mathbf{Z}} \doteq \mathrm{E}\{\mathbf{Z}|\mathbf{X}\}$ and that using the results we have so far we can write this term as

$$\hat{\mathbf{Z}} \doteq \mathrm{E}\{\mathbf{Z}|\mathbf{X}\} = \mathbf{D}\cdot\mathbf{X}\cdot\mathbf{P}_\lambda\,. \tag{2.6}$$

Then, we can rewrite (2.2) as

$$\hat{\mathbf{P}} = \underset{\mathbf{P}\in\mathcal{P}^N}{\operatorname{argmin}}\, ||\mathbf{Y} - \hat{\mathbf{Z}}\cdot\mathbf{P} - \mathbf{1}_\rho\cdot(\boldsymbol{\delta}^{\texttt{MIX}})^T||^2\,. \tag{2.7}$$

Interestingly, removing the constraints $\mathbf{P}\in\mathcal{P}^N$ in (2.7) leads to an estimator which is not only unbiased and asymptotically efficient, as proven in [35], but also makes a detailed performance analysis manageable as we show in Section 2.4. In the rest of this chapter, we focus on the unconstrained estimator and refer to [35] for further information about the constrained variant. The solution to the unconstrained problem, if we assume that $\rho > N$, is given by the Moore-Pensore pseudo-inverse, as shown in [35]:

$$\hat{\mathbf{P}} = (\hat{\mathbf{Z}}^T\hat{\mathbf{Z}})^{-1}\hat{\mathbf{Z}}^T(\mathbf{Y} - \mathbf{1}_\rho\cdot(\boldsymbol{\delta}^{\texttt{MIX}})^T)\,. \tag{2.8}$$

Table 2.2: Notation of our analysis of the binomial pool.

Symb.	Meaning
λ_i	Sending rate of real messages of user i.
δ_i	Sending rate of dummy messages of user i.
α	Prob. that a message inside the binomial pool leaves in a round.
λ'_j	Real message receiving rate $\lambda'_j \doteq \sum_{i=1}^{N} \lambda_i p_{j,i}$.
α_q	Auxiliary parameter $\alpha_q \doteq \alpha/(2-\alpha)$.
α_r	Auxiliary parameter $\alpha_r \doteq \alpha(2-\alpha)/\left(2-\alpha(2-\alpha)\right)$.
$\delta_{\texttt{SEND}}$	Total number of average sender-based dummies, $\delta_{\texttt{SEND}} = \sum_{i=1}^{N} \delta_i$.
$\delta_{\texttt{MIX}}$	Total number of average mix-based dummies, $\delta_{\texttt{MIX}} = \sum_{j=1}^{M} \delta_j^{\texttt{MIX}}$.
$\delta_{\texttt{TOT}}$	Total budget of dummies, $\delta_{\texttt{TOT}} = \delta_{\texttt{SEND}} + \delta_{\texttt{MIX}}$.

2.4. Performance Analysis in a Timed Pool Mix with Dummies

In this section, we assess the performance of the least squares estimator in (2.8) with respect to its profiling accuracy, measured as the Mean Squared Error (MSE) of estimated transition probabilities $p_{j,i}$ representing users' behavior (2.1). We consider the particular case when the anonymous communication channel is a *binomial timed pool mix* [14], and the number of messages sent by the users, as well as the dummies generated by the mix, are Poisson-distributed. In a binomial timed pool mix, the firing condition is a timeout and the batching strategy mandates that individual messages leave the pool with probability α every round, i.e., $d_k = \alpha(1-\alpha)^k$. This scenario can be summarized as

$$\begin{gathered} X^r_{\lambda,i} \sim \text{Poiss}\left(\lambda_i\right), \quad X^r_{\delta,i} \sim \text{Poiss}\left(\delta_i\right), \quad Y^r_{\delta,j} \sim \text{Poiss}\left(\delta_j^{\texttt{MIX}}\right), \\ P_{\lambda_i} = \lambda_i/(\lambda_i+\delta_i), \quad d_k = \alpha(1-\alpha)^k, \end{gathered} \tag{2.9}$$

where λ_i is the *user sending rate*, and δ_i is the *user dummy rate*, representing the average number of real messages, respectively dummies, sent by user i. Even though the results we provide correspond to the above case we must stress that the reasoning followed in the derivation is applicable to any other system that can be represented by the model in Sect. 2.2. Table 2.2 summarizes the new notation introduced in this section.

2.4.1. Profiling Error of the Least Squares Estimator

Under the hypotheses stated in (2.9), in Appendix 2.B we show that the least squares estimator is unbiased and the MSE of a single transition probability

estimated is given by:

$$\mathrm{MSE}_{j,i} \approx \frac{1}{\rho} \cdot \frac{1}{\alpha_q} \cdot \frac{1}{\lambda_i} \cdot \left(1 + \frac{\delta_i}{\lambda_i}\right) \cdot \left(1 - \frac{\lambda_i + \delta_i}{\sum_{k=1}^{N}(\lambda_k + \delta_k)}\right) \cdot \left(\sum_{k=1}^{N} \lambda_k p_{j,k} + \delta_j^{\texttt{MIX}} - \frac{\alpha_q}{\alpha_r} \sum_{k=1}^{N} \lambda_k P_{\lambda_k} p_{j,k}^2\right), \tag{2.10}$$

where $\alpha_q \doteq \dfrac{\alpha}{2-\alpha}$ and $\alpha_r \doteq \dfrac{\alpha(2-\alpha)}{2-\alpha(2-\alpha)}$. This result holds when: i) the probability that each sender sends a message to receiver j is negligible when compared to the rate at which receiver j receives messages from all users ($p_{j,i} \ll \sum_k \lambda_k p_{j,k}$), ii) the number of rounds observed is large enough ($\rho \to \infty$), and iii) $\lambda_i + \delta_i \ll \left(\sum_k (\lambda_k + \delta_k)\right)^2$.

Interestingly, the terms in (2.10) that depend on i and j can be decoupled:

$$\mathrm{MSE}_{j,i} \approx \frac{1}{\rho} \cdot \frac{1}{\alpha_q} \cdot \epsilon_s(i) \cdot \epsilon_r(j)\,. \tag{2.11}$$

where $\epsilon_s(i)$ and $\epsilon_r(j)$ denote functions that only depend on the sender i and the receiver j respectively. This property proves to be very useful when designing strategies to distribute the dummy traffic as we later see in Sect. 2.5.

The latter expression allows to extract qualitative conclusions on the protection dummy traffic offers to senders and receivers. As it was already shown in [80], the MSE decreases with the number of rounds observed as $1/\rho$, and delaying messages in the pool increases the $\mathrm{MSE}_{j,i}$ by a factor $(2-\alpha)/\alpha$ with respect to an scenario with no delay (i.e., $\alpha = 1$).

We now analyze the contribution of the users' behavior to the MSE. The sender-side contribution $\epsilon_s(i)$ consists of three terms:

$$\epsilon_s(i) = \frac{1}{\lambda_i} \cdot \left(1 + \frac{\delta_i}{\lambda_i}\right) \cdot \left(1 - \frac{\lambda_i + \delta_i}{\sum_{k=1}^{N}(\lambda_k + \delta_k)}\right). \tag{2.12}$$

1. The term $1/\lambda_i$ implies that the error when estimating the profile $\mathbf{q}_i = [p_{1,i}, \cdots, p_{M,i}]^T$ decreases as that user participates in the system more often. Naturally, when more information about the user becomes available to the adversary, it becomes easier to accurately estimate her behavior.

2. The second term, $1+\delta_i/\lambda_i$, is always larger or equal than one, meaning that sender-based dummies always hinder the attacker's estimation. The weight of this component depends on the ratio between the dummy rate and the sending rate. Hence, a user who sends real messages very often would need to send a many more dummies to get the same level of protection than a user who rarely participates in the system.

3. The last term is in general negligible since, in a normal scenario, the participation of a single user is negigible when compared to the total traffic, i.e., $\lambda_i + \delta_i \ll \sum_{k=1}^{N}(\lambda_k + \delta_k)$. However, when user i's traffic is clearly dominant among the others, this term decreases the overall gain i gets from dummies. Therefore, although sender-based dummies always increase the protection of a user, they offer diminishing returns when only one user is trying to protect herself by sending dummies.

On the other hand, the receiver-side contribution, $\epsilon_r(j)$, consists of three summands:

$$\epsilon_r(j) = \sum_{k=1}^{N} \lambda_k p_{j,k} + \delta_j^{\texttt{MIX}} - \frac{\alpha_q}{\alpha_r} \sum_{k=1}^{N} \lambda_k P_{\lambda_k} p_{j,k}^2 \,. \tag{2.13}$$

1. The first summand is the rate at which j receives real messages from the senders. We call this term *receiver rate* and denote it by λ'_j. It is interesting to note that, contrary to the sending rates where large values of λ_i compromise the anonymity of the senders; large values of receiver rates increase the protection of the receivers. In other words, it is harder for the attacker to estimate probabilities related to a receiver which is contacted by a large number of senders than related to one receiving few messages.

2. The second summand is the rate at which j receives dummy messages from the mix. The interesting part about this summand is that it can be adjusted by the mix, to give more protection to a specific receiver j by increasing the number of dummies addressed to that recipient.

3. The last summand depends on the mix parameters and the users' behavior. Since $\alpha_q/\alpha_r \leq 1$ and $P_{\lambda_k} \leq 1$, when users do not focus their messages in few others, i.e., $p_{j,i} \ll 1$, this summand becomes negligible. However, if there is no dummy traffic ($P_{\lambda_k} = 1$ and $\delta_j^{\texttt{MIX}} = 0$) and no pool is implemented ($\alpha_q/\alpha_r = 1$), this term must be taken into account. In this case $\epsilon_r(j)$ depends on the variance of the outputs, i.e. $\sum_{k=1}^{N} \lambda_k p_{j,k}(1 - p_{j,k})$, meaning that it would easier for the attacker to estimate probabilities $p_{j,k}$ of receivers that get messages from senders whose behavior has low variance (i.e., senders that always choose the same receiver, $p_{j,k} = 1$, or users that never send to a receiver, $p_{j,k} = 0$). Adding delay or introducing dummy traffic increases the variance of the output, thus reducing the dependency of the error on the sending profiles.

The fact that we can differentiate the contribution of i and j in (2.10) also allows for a graphic interpretation of the adversary's estimation error. Figure 2.2a represents the values of $\text{MSE}_{j,i}$ as a function of i and j, in an scenario without dummies where for simplicity we have assumed that the sending rates are

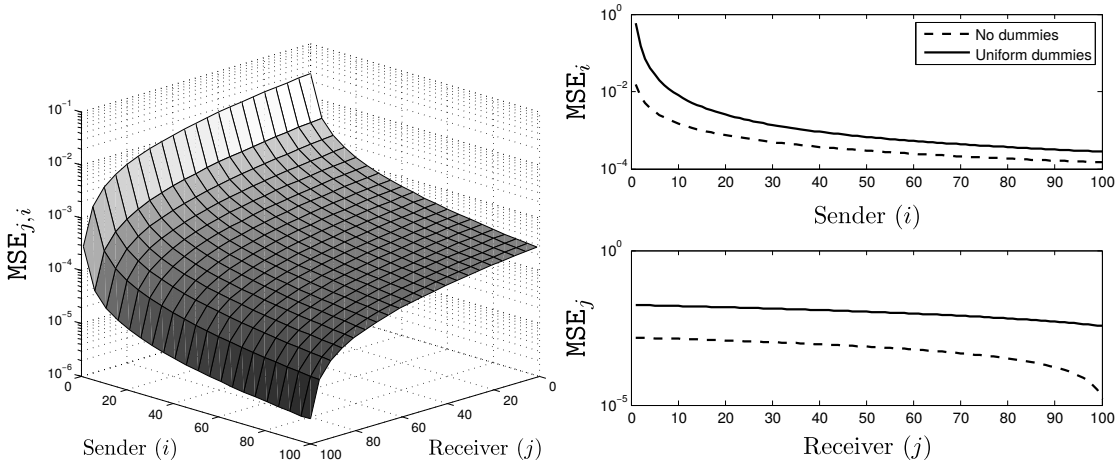


Figure 2.2: (a) $\text{MSE}_{j,i}$ as a function of i and j in an scenario where λ_i are sorted in ascending order and λ'_j in descending order. (b) Comparison of the average $\text{MSE}_{j,i}$ along j and i with and without dummies. ($N = 100$, $M = 100$, $\rho = 10\,000$, $\alpha = 0.5$, $\sum \lambda_k = 500$. In (b), $\delta_{\texttt{SEND}} = \delta_{\texttt{MIX}} = 250$).

distributed in ascending order according to the senders' index i, and the receiving rates are distributed in descending order according to the receivers' index j. Fig. 2.2b shows the average $\text{MSE}_{j,i}$ over j and i, offering a comparison with a system where the distribution of the dummies is uniform in both the input and output flows: $\epsilon_s(i)$ determines the evolution of $\text{MSE}_{j,i}$ with i (top) and $\epsilon_r(j)$ the evolution with j (bottom). This means that by distributing dummies among sender-based and mix-based dummies, which in turn modify the value of $\epsilon_s(i)$ and $\epsilon_r(j)$, we can shape the $\text{MSE}_{j,i}$. We use this idea in the next section to design dummy strategies that satisfy different privacy criteria.

2.5. Designing Dummy Traffic Strategies

In this section, we study how to distribute dummy traffic in order to guarantee different privacy criteria. In other words, we aim at finding the values of the parameters δ_i for $i \in \{1, \cdots, N\}$ and $\delta_j^{\texttt{MIX}}$ for $j \in \{1, \cdots, M\}$ that maximize a certain cost function representing some privacy objective. We assume that the *total number of dummies* δ_{TOT} that can be sent *on average* per round is constrained. We denote the average number of sender-based dummies on each round as $\delta_{\texttt{SEND}} \doteq \sum_{i=1}^{N} \delta_i$, and the average number of mix-based dummies as $\delta_{\texttt{MIX}} \doteq \sum_{j=1}^{M} \delta_j^{\texttt{MIX}}$. We put no restriction on the distribution of dummies among senders and mix other than $\delta_{\texttt{SEND}} + \delta_{\texttt{MIX}} \leq \delta_{\texttt{TOT}}$. For notational simplicity, in the remainder of the section we omit the constraints $\delta_i, \delta_j^{\texttt{MIX}} \geq 0$, $\sum_{i=1}^{N} \delta_i = \delta_{\texttt{SEND}}$ and $\sum_{j=1}^{M} \delta_j^{\texttt{MIX}} = \delta_{\texttt{MIX}}$ in the equations.

In order to keep the optimization problems tractable, we assume that the contribution of a single user to the total input traffic is negligible (i.e., $\lambda_i + \delta_i \ll \sum_{k=1}^{N}(\lambda_k + \delta_k)$) and that users do not focus their traffic in a specific receiver (i.e., $p_{j,i} \ll 1$). In this case, we can approximate (2.10) as:

$$\widetilde{\mathtt{MSE}}_{j,i} = \frac{1}{\rho} \cdot \frac{1}{\alpha_q} \cdot \frac{1}{\lambda_i} \cdot \left(1 + \frac{\delta_i}{\lambda_i}\right) \cdot \left(\lambda'_j + \delta_j^{\mathtt{MIX}}\right) = \frac{1}{\rho} \cdot \frac{1}{\alpha_q} \cdot \tilde{\epsilon}_s(i) \cdot \tilde{\epsilon}_r(j)\,, \tag{2.14}$$

where $\lambda'_j \doteq \sum_{k=1}^{N} \lambda_k p_{j,k}$ is the receiver rate of j.

2.5.1. Increase the Protection by a Multiplicative Factor

In this section, we design a dummy strategy that, given a budget of dummies $\delta_{\mathtt{TOT}}$, increases $\mathrm{MSE}_{j,i}$ of *each* transition probability $p_{j,i}$ by a factor $\beta \geq 1$ as large as possible with respect to the MSE when there are no dummies, denoted by $\mathrm{MSE}^0_{j,i}$. Departing from (2.14), we can formalize this problem as:

$$\begin{aligned} \underset{\delta_i, \delta_j^{\mathtt{MIX}},\ \forall i,j}{\text{maximize}} \quad & \beta \\ \text{subject to} \quad & \widetilde{\mathtt{MSE}}_{j,i} \geq \beta \cdot \widetilde{\mathtt{MSE}}^0_{j,i}, \qquad \forall i,j \\ & \delta_{\mathtt{SEND}} + \delta_{\mathtt{MIX}} = \delta_{\mathtt{TOT}}\,. \end{aligned} \tag{2.15}$$

Note that β is independent of i, j, i.e., we want to increase the estimation error of each $p_{j,i}$, at least, by the same factor. Since the effects of the sender-based and mix-based dummies can be decoupled, we can decouple the increase factor as $\beta = \beta_{\mathtt{SEND}} \cdot \beta_{\mathtt{MIX}}$ and then split the optimization into three subproblems:

1. Distribute $\delta_{\mathtt{SEND}}$ among each δ_i to increase $\tilde{\epsilon}_s(i)$ by a factor $\beta_{\mathtt{SEND}}$ for all i.
2. Distribute $\delta_{\mathtt{MIX}}$ among each $\delta_j^{\mathtt{MIX}}$ to increase $\tilde{\epsilon}_r(j)$ by a factor $\beta_{\mathtt{MIX}}$ for all j.
3. Distribute $\delta_{\mathtt{TOT}}$ between $\delta_{\mathtt{SEND}}$ and $\delta_{\mathtt{MIX}}$ to maximize the overall increase $\beta = \beta_{\mathtt{SEND}} \cdot \beta_{\mathtt{MIX}}$.

Optimal distribution of sender-based dummies. We want to find the values of δ_i that maximize the factor $\beta_{\mathtt{SEND}}$ by which every $\tilde{\epsilon}_s(i)$ increases. Since $\tilde{\epsilon}_s(i) = \lambda_i^{-1}(1 + \delta_i/\lambda_i)$, sending δ_i dummies increases the MSE for sender i by a factor of $1 + \delta_i/\lambda_i$. Since there is a total budget of sender-based dummies shared among all senders $\sum_{i=1}^{N} \delta_i = \delta_{\mathtt{SEND}}$, the optimal strategy will increase *each* $\tilde{\epsilon}_s(i)$ exactly by the same factor $\beta_{\mathtt{SEND}} = 1 + \delta_i/\lambda_i$ (allocating extra dummies to a specific sender k

to make $\tilde{\epsilon}_s(k)$ strictly larger than $\beta_{\texttt{SEND}} \cdot 1/\lambda_k$ does not help towards maximizing $\beta_{\texttt{SEND}}$). By combining $\beta_{\texttt{SEND}} = 1 + \delta_i/\lambda_i$ and $\sum_{i=1}^{N} \delta_i = \delta_{\texttt{SEND}}$, we obtain:

$$\beta_{\texttt{SEND}} = 1 + \frac{\delta_{\texttt{SEND}}}{\sum_{k=1}^{N} \lambda_k} \Longrightarrow \delta_i = \frac{\lambda_i}{\sum_{k=1}^{N} \lambda_k} \cdot \delta_{\texttt{SEND}}, \qquad \forall i\,. \tag{2.16}$$

This confirms the intuition given in Sect. 2.4, that the number of dummies a user should send to achieve a certain level of protection is proportional to her sending rate of real messages.

Optimal distribution of mix-based dummies. Similarly, we want to find the values of $\delta_j^{\texttt{MIX}}$ that increase $\tilde{\epsilon}_r(j)$ by a factor $\beta_{\texttt{MIX}}$ compared to the dummy-free case. Since $\tilde{\epsilon}_r(j) = \lambda'_j + \delta_j^{\texttt{MIX}}$, assigning a rate $\delta_j^{\texttt{MIX}}$ to receiver j increases the $\widetilde{\texttt{MSE}}_{j,i}$ by a factor of $1 + \delta_j^{\texttt{MIX}}/\lambda'_j$. As in the sender case above, the optimal solution will allocate dummies to each recipient such that exactly $\beta_{\texttt{MIX}} = 1 + \delta_j^{\texttt{MIX}}/\lambda'_j$ for all j. We can now obtain the sender-based dummy distribution, ensuring that $\sum_{j=1}^{M} \delta_j^{\texttt{MIX}} = \delta_{\texttt{MIX}}$, as follows:

$$\beta_{\texttt{MIX}} = 1 + \frac{\delta_{\texttt{MIX}}}{\sum_{m=1}^{M} \lambda'_m} \Longrightarrow \delta_j^{\texttt{MIX}} = \frac{\lambda'_j}{\sum_{m=1}^{M} \lambda'_m} \cdot \delta_{\texttt{MIX}}, \qquad \forall j\,. \tag{2.17}$$

As said in Sect. 2.4, the protection that receivers enjoy is proportional to their receiving rate. Therefore, to increase all $\text{MSE}_{j,i}$s by the same factor, more mix-based dummies have to be given to those receivers that receive more real messages.

Optimal distribution of the overall amount of dummies. Using the distributions obtained, and since $\sum_{k=1}^{N} \lambda_k = \sum_{m=1}^{M} \lambda'_m$, we can write $\widetilde{\texttt{MSE}}_{j,i}$ as

$$\widetilde{\texttt{MSE}}_{j,i} = \widetilde{\texttt{MSE}}^{0}_{j,i} \cdot \beta_{\texttt{SEND}} \cdot \beta_{\texttt{MIX}} = \widetilde{\texttt{MSE}}^{0}_{j,i} \left(1 + \frac{\delta_{\texttt{SEND}}}{\sum_{k=1}^{N} \lambda_k}\right) \left(1 + \frac{\delta_{\texttt{MIX}}}{\sum_{k=1}^{N} \lambda_k}\right). \tag{2.18}$$

The distribution of the total amount of dummies that maximizes the increase in $\widetilde{\texttt{MSE}}_{j,i}$ is therefore $\delta_{\texttt{SEND}} = \delta_{\texttt{MIX}} = \delta_{\texttt{TOT}}/2$. This result is particularly interesting: if we are to increase the relative protection of each user equally, the protection we get from sender-based and mix-based dummies is the same regardless of the system parameters. That is, assigning all our available dummies to the senders or to the mix is equivalent in terms of MSE, and distributing the dummies evenly between the input and output flow is optimal, being the maximum achievable gain $\beta \approx \left(1 + \frac{\delta_{\texttt{TOT}}/2}{\sum_k \lambda_k}\right)^2$.

2.5.2. Maximize the Minimum Protection of all Relations

Our second design strategy aims at ensuring that the minimum level of protection of *all sender-receiver relationships* in the system is as large as possible. This

implies that dummies are assigned to senders i and receivers j in relationships whose estimation error $\text{MSE}_{j,i}$ is low, in order to increase the minimum $\text{MSE}_{j,i}$ in the system. From a graphical point of view, we can see this as a two-dimensional waterfilling problem: we need to increase the lower $\text{MSE}_{j,i}$ in Fig. 2.2a up to a minimum, which can be larger as more dummies $\delta_{\texttt{TOT}}$ are available. More formally, we want to solve:

$$\begin{aligned} &\underset{\delta_i, \delta_j^{\texttt{MIX}},\ \forall i,j}{\text{maximize}} && \min_{i,j} \widetilde{\texttt{MSE}}_{j,i} \\ &\text{subject to} && \delta_{\texttt{SEND}} + \delta_{\texttt{MIX}} = \delta_{\texttt{TOT}}\,. \end{aligned} \tag{2.19}$$

As in the previous problem, we can separate the problem in three steps:

1. Distribute $\delta_{\texttt{SEND}}$ among the δ_i to maximize $\min_i \tilde{\epsilon}_s(i)$.
2. Distribute $\delta_{\texttt{MIX}}$ among the $\delta_j^{\texttt{MIX}}$ to maximize $\min_j \tilde{\epsilon}_r(j)$.
3. Distribute $\delta_{\texttt{TOT}}$ between $\delta_{\texttt{SEND}}$ and $\delta_{\texttt{MIX}}$ to maximize the minimum $\widetilde{\texttt{MSE}}_{j,i}$ in the system.

Optimal distribution of sender-based dummies. We aim at finding the values of δ_i that increase the minimum value of $\tilde{\epsilon}_s(i) = \frac{1}{\lambda_i}\left(1 + \frac{\delta_i}{\lambda_i}\right)$ over i, making it as large as possible given the budget of dummies. This subproblem can be formulated as

$$\begin{aligned} &\underset{\delta_i,\ \forall i}{\text{maximize}} && \min_i \tilde{\epsilon}_s(i) \\ &\text{subject to} && \sum_{i=1}^{N} \delta_i = \delta_{\texttt{SEND}}\,. \end{aligned} \tag{2.20}$$

Let $\mathcal{A}$ be the set containing the indices of those senders to whom we assign dummies, i.e., $\mathcal{A} \doteq \{i : \delta_i > 0\}$. Let $\tilde{\epsilon}_{s,\texttt{MIN}}$ be the minimum value of $\tilde{\epsilon}_s(i)$ we achieve with this strategy. Then, the following statements are true:

- We do not assign sender-based dummies to those users k whose $\tilde{\epsilon}_s(k) \geq \tilde{\epsilon}_{s,\texttt{MIN}}$ without dummies; i.e., we only use sender-based dummies to help users achieve that minimum.
- There is no gain in assigning dummies to a user k if by doing so we are increasing $\tilde{\epsilon}_s(k)$ above any other $\tilde{\epsilon}_s(i)$; i.e., every user $k \in \mathcal{A}$ fullfills $\tilde{\epsilon}_s(k) = \tilde{\epsilon}_{s,\texttt{MIN}}$.

Given $\tilde{\epsilon}_s(k) = \tilde{\epsilon}_{s,\texttt{MIN}}$, and to ensure $\sum_{k=1}^{N} \delta_k = \sum_{k\in\mathcal{A}} \delta_k = \delta_{\texttt{SEND}}$ we can get an expression for $\tilde{\epsilon}_{s,\texttt{MIN}}$:

$$\tilde{\epsilon}_{s,\texttt{MIN}} = \frac{1}{\lambda_k}\left(1 + \frac{\delta_k}{\lambda_k}\right) \Longrightarrow \tilde{\epsilon}_{s,\texttt{MIN}} = \frac{\delta_{\texttt{SEND}} + \sum_{k\in\mathcal{A}} \lambda_k}{\sum_{k\in\mathcal{A}} \lambda_k^2}\,. \tag{2.21}$$

In order to compute $\mathcal{A}$, we assume w.l.o.g. that the indices are given to users such that their sending frequencies are sorted in *descending* order, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$ and we let $\mathcal{A}_i \doteq \{1, 2, \cdots, i\}$. Then, $\mathcal{A} = \mathcal{A}_n$ where n is the minimum value that meets[1]

$$\frac{1}{\lambda_n} \leq \frac{\delta_{\texttt{SEND}} + \sum_{k \in \mathcal{A}_n} \lambda_k}{\sum_{k \in \mathcal{A}_n} \lambda_k^2} \leq \frac{1}{\lambda_{n+1}} \,. \tag{2.22}$$

Finally, we assign

$$\delta_i = \begin{cases} \lambda_i \left(\lambda_i \tilde{\epsilon}_{s,\texttt{MIN}} - 1\right), & \text{if } i \in \mathcal{A}_n \,, \\ 0, & \text{otherwise.} \end{cases} \tag{2.23}$$

Optimal distribution of mix-based dummies. Similarly, we aim at finding the values of $\delta_j^{\texttt{MIX}}$ that increase the minimum value of $\tilde{\epsilon}_r(j)$, making it as large as possible given the budget of dummies. The problem can be formulated as:

$$\begin{aligned} &\underset{\delta_j^{\texttt{MIX}},\ \forall j}{\text{maximize}} \quad \min_j \tilde{\epsilon}_r(j) \\ &\text{subject to} \quad \sum_{j=1}^{M} \delta_j^{\texttt{MIX}} = \delta_{\texttt{MIX}} \,, \end{aligned} \tag{2.24}$$

where $\tilde{\epsilon}_r(j) = \lambda'_j + \delta_j^{\texttt{MIX}}$.

We define the set $\mathcal{B}$ as the send of receivers that get mix-based dummies, $\mathcal{B} \doteq \{j : \delta_j^{\texttt{MIX}} > 0\}$ and the minimum value of our optimization function we achieve with this strategy as $\tilde{\epsilon}_{r,\texttt{MIN}}$. Then, following the procedure described above, we get

$$\tilde{\epsilon}_{r,\texttt{MIN}} = \frac{\delta_{\texttt{MIX}} + \sum_{j \in \mathcal{B}} \lambda'_j}{|\mathcal{B}|} \,, \tag{2.25}$$

where $|\mathcal{B}|$ denotes the number of elements of $\mathcal{B}$. If the receiver rates are sorted in *ascending* order, $\lambda'_1 \leq \lambda'_2 \leq \cdots \leq \lambda'_M$ and $\mathcal{B}_j \doteq \{1, 2, \cdots, j\}$, then the set of receivers that receive dummy messages is $\mathcal{B} = \mathcal{B}_n$ where the value of n is the smallest that meets

$$\lambda'_n \leq \frac{\delta_{\texttt{MIX}} + \sum_{j \in \mathcal{B}_n} \lambda'_j}{|\mathcal{B}_n|} \leq \lambda'_{n+1} \,. \tag{2.26}$$

Finally, we assign

$$\delta_j^{\texttt{MIX}} = \begin{cases} \tilde{\epsilon}_{r,\texttt{MIN}} - \lambda'_j, & \text{if } j \in \mathcal{B}_n, \\ 0, & \text{otherwise.} \end{cases} \tag{2.27}$$

[1] If the condition is not met because all $1/\lambda_n \leq \tilde{\epsilon}_{s,\texttt{MIN}}(\mathcal{A}_n)$, then we can assume that $n = N$, i.e., all users will send dummies.

Optimal distribution of the overall amount of dummies. In this case we cannot get a closed-form expression for the optimal distribution of $\delta_{\texttt{TOT}}$ among $\delta_{\texttt{SEND}}$ and $\delta_{\texttt{MIX}}$, since it depends on the sizes of the sets $\mathcal{A}$ and $\mathcal{B}$. The minimum $\widetilde{\texttt{MSE}}_{j,i}$ we achieve is for relationships where both sender and receiver are allocated dummies, i.e., $i \in \mathcal{A}$ and $j \in \mathcal{B}$. Hence we can obtain this minimum by plugging the distributions (2.23) and (2.27) into (2.14), obtaining

$$\min_{j,i} \widetilde{\texttt{MSE}} = \frac{1}{\rho} \cdot \frac{1}{\alpha_q} \cdot \frac{\delta_{\texttt{SEND}} + \sum_{k \in \mathcal{A}} \lambda_k}{\sum_{k \in \mathcal{A}} \lambda_k^2} \cdot \frac{\delta_{\texttt{MIX}} + \sum_{m \in \mathcal{B}} \lambda'_m}{|\mathcal{B}|} \,. \tag{2.28}$$

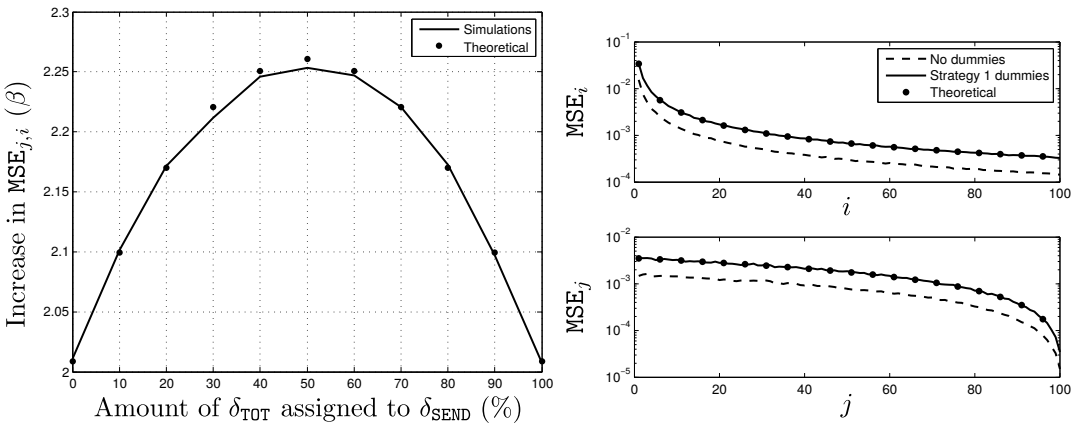
Optimal values for $\delta_{\texttt{SEND}}$ and $\delta_{\texttt{MIX}}$ can be computed by performing an exhaustive search along $\delta_{\texttt{SEND}} + \delta_{\texttt{MIX}} = \delta_{\texttt{TOT}}$, computing each time the sets $\mathcal{A}$ and $\mathcal{B}$ as explained above. It is interesting to note that, if the number of dummies available is large enough, i.e., $\delta_{\texttt{TOT}} \to \infty$, every sender and receiver is assigned dummies. In this case, since $\sum_{k=1}^{N} \lambda_k = \sum_{m=1}^{M} \lambda'_m$, the optimal strategy would be to distribute the total amount of dummies evenly between the input and the output traffics, i.e., $\delta_{\texttt{SEND}} = \delta_{\texttt{MIX}} = \delta_{\texttt{TOT}}/2$.

2.6. Empirical Evaluation

In this section we evaluate the performance of the dummy traffic design strategies designed in Sect. 2.5, and validate them against the theoretical bound for the adversary's error in (2.10) through a simulator written in the Matlab language.[2] The scope of this analysis is focused on supporting our theoretical findings rather than comparing our estimator with existing attacks. The only attack in the literature extended to cover dummy traffic is the Statistical Disclosure Attack (SDA) [5, 21] and it is already shown in [34, 80] that the least squares-based approach performs asymptotically better than SDA. It must be noted that the Bayesian inference estimator (Vida) in [32] may return a better estimation than our least squares estimator. However, its computational cost is huge even for a threshold mix [80] and it would become prohibitive in a pool mix with dummies.

Experimental Setup. We simulate a system with $N = 100$ senders and $M = 100$ receivers. The sending frequencies of the users are sorted in ascending order, in such a way that λ_i is proportional to i, and the average total number of real messages sent by all users is $\sum \lambda_i = 500$. The sending profiles $\mathbf{q}_i$ are set such that user i sends messages to herself and all other users $k < i$ with the same probability, i.e., $p_{j,i} = 1/i$ if $j \leq i$ and $p_{j,i} = 0$. This ensures that receiving rates λ'_j are sorted in descending order. The probability that a message is flushed from the pool after each round is set to $\alpha = 0.5$, and the number of rounds observed by the attacker is $\rho = 10\,000$. The theoretical $\text{MSE}_{j,i}$ for this scenario without

[2] The code will be available upon request.



(a) (b)

Figure 2.3: (a) Evolution of β with the fraction of dummies distributed among senders and mix. (b) Average $\text{MSE}_{j,i}$ evolution over i (top) and j (bottom) when dummies are distributed uniformly among senders and mix. ($N = 100$, $M = 100$, $\rho = 10\,000$, $\alpha = 0.5$, $\delta_{\texttt{TOT}} = 500$)

dummies is shown in Fig. 2.2a. Though not realistic, this experiment is sufficient to illustrate the operation of the strategies in Sect. 2.5. The amount of dummies that users and mix send and their distribution change between experiments. We run four experiments, two for each dummy strategy in Sect. 2.5. We repeat each experiment 200 times and plot the average results.

2.6.1. Increase the Protection by a Multiplicative Factor β

First, we study the influence of the distribution of dummies among senders and mix in the factor β that can be achieved with this strategy, when on average $\delta_{\texttt{TOT}} = 500$ dummies per round are available. Figure 2.3a shows the evolution of β for different distributions of dummy messages between senders ($\delta_{\texttt{SEND}}$) and mix ($\delta_{\texttt{MIX}}$). We see that the maximum increase is achieved when dummies are divided equally between the senders and the mix, as predicted in Sect. 2.5.1. Also, this optimal factor is close to the value we predicted $\beta \approx \left(1 + \frac{\delta_{\texttt{TOT}}/2}{\sum_k \lambda_k}\right)^2 = 2.25$.

For the particular case where $\delta_{\texttt{SEND}} = \delta_{\texttt{MIX}} = \delta_{\texttt{TOT}}/2$, we plot in Fig. 2.3b the average $\text{MSE}_{j,i}$ over i (top) and j (bottom) with and without dummies (note the vertical axis logarithmic scale). We see that indeed all $\text{MSE}_{j,i}$ increase by a constant factor, $\beta = 2.261$. The figure also shows that (2.10) accurately models the profiling error.

2.6.2. Maximize the Minimum Protection of all Relations

First, we study the influence of the distribution of dummies among senders and mix on the maximum minimum $\text{MSE}_{j,i}$ that can be achieved with this strategy, when on average $\delta_{\texttt{TOT}} = 500$ dummies per round are available. Fig. 2.4a shows the evolution of the average minimum $\text{MSE}_{j,i}$ depending on the distribution of dummies between the senders and the mix. In the scenario considered in our experiment, the maximum minimum $\text{MSE}_{j,i}$ achievable is obtained when approximately 40% of the dummies are assigned to the senders and the remaining 60% to the mix. This is because, in this strategy, the rate of sender-based dummies depends quadratically on the real sending rate (c.f. (2.23)), while the number of mix-based dummies depends linearly on the real receiving rate (c.f. (2.27)). Hence, mix-based dummies can be distributed more efficiently and it is preferable to assign the mix a larger budget than to the senders. We note that this result depends strongly on the users behavior. In fact, if the real traffic is distributed uniformly among receivers but few senders generate the majority of the traffic, allocating a large fraction of dummy traffic to the senders becomes the best option.

This is better shown in Fig. 2.4b. The top plot shows the $\text{MSE}_{j,i}$ along i when there are no dummies, and when only sender-based dummies are available ($\delta_{\texttt{SEND}} = \delta_{\texttt{TOT}}$; $\delta_{\texttt{MIX}} = 0$). As expected, more dummies increase the minimum $\text{MSE}_{j,i}$, but, since the average number of sender-based dummies depends quadratically on the real sending rate, few senders with high rates exhaust the budget, which constrains the maximum minimum error achievable in the system. On the other hand, allocating all the dummies to the mix (Fig. 2.4b, bottom) allows to spread the distribution of dummies among more relationships, which in turn provides better overall protection than the previous case.

2.7. Discussion

In this section we discuss how to adapt the derivation of the least squares estimator in Sect. 2.3 to scenarios where pool and users' behavior are outside of the model considered throughout the chapter.

Non-stationary delay characteristic. Our findings can be easily extended to non-stationary pool mixes, whose delay characteristic changes in each round. In this case, the delay characteristic is no longer denoted by the parameters d_k, but by a two-parameter function $F_{r,k}$ that represents the probability that a message that arrives to the mix in round k leaves in round r. With this parameter, it is easy to build a new matrix $\mathbf{D}$ for the least-squares estimator (2.8). For the full details, we refer to the original paper [81] where we used this variation to derive the attack.

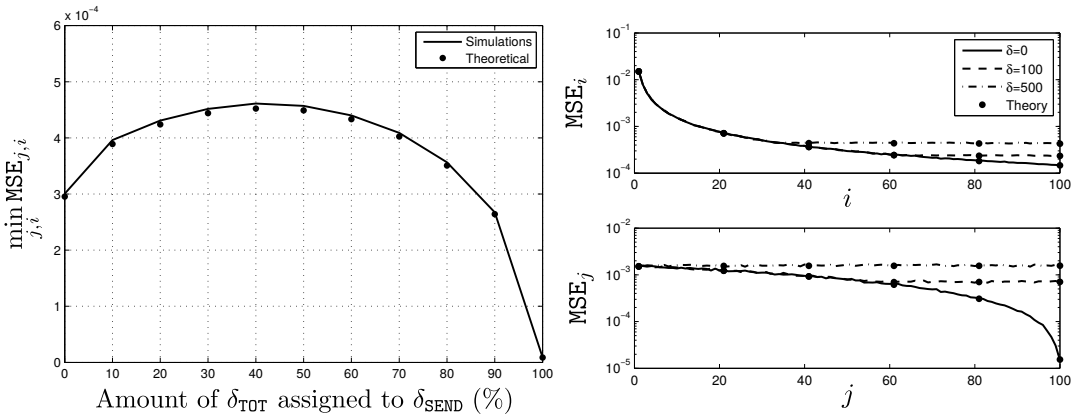


Figure 2.4: (a) Evolution of the minimum $\mathrm{MSE}_{j,i}$ with the fraction of dummies distributed among senders and mix. (b) Average $\mathrm{MSE}_{j,i}$ evolution over i when only sender-based dummies are available (top), and j when only mix-based dummies are available (bottom). ($N = 100$, $M = 100$, $\rho = 10\,000$, $\alpha = 0.5$, $\delta_{\mathtt{TOT}} = 100, 500$)

Non-stationary sending profiles. In practice, users' behavior is expected to change over time. Our estimator can be adapted to account for dynamic profiles by implementing the Recursive Least Squares algorithm [82]. This algorithm includes a *forgetting factor*, which determines how fast the algorithm "forgets" past observations. Tuning this parameter, one can choose between getting a high-variance estimator of the recent users' sending profile or obtaining a more stable long-term sending profile.

Non-independent users with memory. Although our model considers disjoint sets of senders and receivers, it can easily accommodate the case where users both send and receive messages. In this scenario, users' sending behavior may be dependent on messages sent or received in the past (e.g., email replies). Given a model of these interactions between users one can compute the expected value of the output observations given the inputs, and then proceed with the derivation of the estimator as in Sect. 2.3.

Non-stationary dummy strategies. If the probability of sending a real message (P_{λ_i}) changes over time, a per-round probability $P^r_{\lambda_i}$ could be defined. This dynamic probability can be used in the derivations in the Appendix (c.f. (2.30)) to account for the effect of this variation on the attacker's estimation of the hidden variables z^r_i. When the average mix-based dummies ($\delta^{\mathtt{MIX}}_j$) vary over time, an aware attacker can include this behavior in (2.3), modifying the expected value of the outputs and thus the attack.

Complex batching strategies. Our anonymous channel model does not cover

pool mixes whose batching strategy depends on the number of messages in the pool, such as that used by Mixmaster [4]. However, extending our model to this scenario is straightforward: the adversary can estimate the average number of messages in the pool by discarding a percentage of the incoming messages that are expected to be dummy, and therefore she can get an estimate of the average number of messages from each user that leave in each round, z_i^r. The estimator would still be formulated as (2.8).

2.8. Conclusions

In this chapter, we have proposed a methodology to analyze mix-based anonymous communication systems with dummy traffic. Following a least squares approach, we derive an estimator of the probability that a user sends messages to a receiver. This estimator allows us to characterize the error of the adversary when recovering user profiles, or individual probabilities, with respect to the system parameters. Furthermore, it can be used to design dummy strategies that satisfy a wide range of privacy criteria.

As an example, we have studied the performance of the least squares estimator on a timed binomial pool mix, which enables us to derive qualitative conclusions about the effects of dummy traffic on the adversary's error. We have used this estimator to design dummy strategies that, given a budget of dummies, achieve two privacy targets: increase the protection of each sender and receiver relationship equally, and maximize the minimum protection provided to any relationship between users. The empirical evaluation of these strategies validates our theoretical results and confirms the qualitative intuitions drawn in the performance analysis.

Our methodology improves our understanding on the effect of dummy traffic on privacy in anonymous communication systems. It can be seen as a step forward towards the development of a systematic method do design dummy traffic, especially important to evaluate and improve privacy protection in deployed mix-based systems such as [4, 5].

Appendix

2.A. Conditional Expectations in Our Model

We compute the expressions of different conditional expectations in our pool mix model introduced in Section 2.2.

Expectation of $Y_{j,i}^r$ given Z. Note that the messages from sender i that leave the mix in round r, i.e., z_i^r, are sent to receiver j, each one, with probability $p_{j,i}$. Mathematically, we can model $Y_{j,i}^r$ given $\mathbf{Z}$ as a binomial distribution:

$$Y_{j,i}^r|\mathbf{Z} \sim \mathrm{Bi}(z_i, p_{j,i})\,, \tag{2.29}$$

Therefore, it is straightforward that $\mathrm{E}\left\{Y_{j,i}^r|\mathbf{Z}\right\} = z_i^r \cdot p_{j,i}$.

Expectation of Z_i^r given X. For simplicity, we assume that, by the time the adversary starts observing the system, the pool is empty. In practice, the initial messages in the pool would appear as noise in the initial output observations and its effect can be disregarded when the number of observations is large, as explained in [35]. The messages sent by user i in round r, i.e., x_i^r, are each real with probability P_{λ_i} or dummy otherwise. Mathematically,

$$X_{\lambda,i}^r|\mathbf{X} \sim \mathrm{Bi}(x_i^r, P_{\lambda_i})\,. \tag{2.30}$$

The real messages go into the pool, and each one waits a number of rounds k randomly chosen following the delay characteristic d_k. The (real) messages from sender i that leave in round r, i.e., Z_i^r, might have been sent in any previous round $s \geq r$. Let $Z_i^{s\to r}$ denote the random variable that models the number of messages from sender i that entered the pool in round s and leave in round r. Note that $Z_i^r = \sum_{s=1}^r Z_i^{s\to r}$. Since the $x_{\lambda,i}^r$ messages that enter the pool in round r might leave in any of the current or following rounds, we can model

$$\{Z_i^{r\to r}, Z_i^{r\to r+1}, Z_i^{r\to r+2}, \cdots | x_{\lambda,i}^r\} \sim \mathrm{Multi}(x_{\lambda,i}^r, \{d_0, d_1, d_2, \cdots\})\,, \tag{2.31}$$

Table 2.3: Additional notation in Appendix 2.B.

Symbol	Meaning
$\mathbf{e}_j$	Error vector in the estimation of $\mathbf{p}_j$, i.e., $\mathbf{e}_j \doteq \hat{\mathbf{p}}_j - \mathbf{p}_j$.
$\mathbf{C}_{e_j}$	Covariance matrix of $\mathbf{e}_j$, i.e., $\mathbf{C}_{e_j} \doteq \mathrm{E}\left\{\mathbf{e}_j \mathbf{e}_j^T\right\}$.
$\mathbf{\Sigma}_{\mathbf{y}_j\|\mathbf{X}}$	Covariance matrix of $\mathbf{y}_j$ given $\mathbf{X}$, shown in (2.38).
$\mathbf{R}_{xx}$	Autocorrelation matrix of the input process, $\mathbf{R}_{xx} \doteq \mathrm{E}\left\{\mathbf{X}^T\mathbf{D}^T\mathbf{D}\mathbf{X}\right\}$.
$\mathbf{F}_\lambda$	Diagonal matrix with the real message sending rates $\lambda_1, \ldots, \lambda_N$.
$\mathbf{F}_\delta$	Diagonal matrix with the dummy message sending rates $\delta_1, \ldots, \delta_N$.
$\mathbf{P}_j j$	Diagonal matrix with the transition probabilities $p_{j,1}, \ldots, p_{j,N}$.
λ'_j	Real message receiving rate $\lambda'_j \doteq \sum_{i=1}^N \lambda_i p_{j,i}$.
λ''_j	Auxiliary parameter $\lambda''_j \doteq \sum_{i=1}^N \lambda_i P_{\lambda_i} p_{j,i}^2$.
α_q	Auxiliary parameter $\alpha_q \doteq \alpha/(2-\alpha)$.
α_r	Auxiliary parameter $\alpha_r \doteq \alpha(2-\alpha)/\left(2-\alpha(2-\alpha)\right)$.
α_s	Auxiliary parameter $\alpha_s \doteq \alpha^3/\left(1-(1-\alpha)^3\right)$.

where Multi stands for multinomial distribution. Finally, using (2.30) and (2.31), we can write

$$\begin{aligned} \mathrm{E}\left\{Z_i^r|\mathbf{X}\right\} &= \textstyle\sum_{s=1}^r \mathrm{E}\left\{Z_i^{s\to r}|\mathbf{X}\right\} \\ &= \textstyle\sum_{s=1}^r \mathrm{E}\left\{\mathrm{E}\left\{Z_i^{s\to r}|X_{\lambda,i}^s\right\} \mid \mathbf{X}\right\} \\ &= \textstyle\sum_{s=1}^r \mathrm{E}\left\{X_{\lambda,i}^s|\mathbf{X}\right\} \cdot d_{r-s} \\ &= \textstyle\sum_{s=1}^r x_i^s \cdot d_{r-s} \cdot P_{\lambda_i}\,, \end{aligned} \tag{2.32}$$

which concludes the derivations.

Expectation of $Y_{j,i}^r$ given X. Using the law of total expectation together (2.29) and (2.32), it is straightforward to get

$$\mathrm{E}\left\{Y_{j,i}^r|\mathbf{X}\right\} = \mathrm{E}\left\{\mathrm{E}\left\{Y_{j,i}^r|\mathbf{Z}\right\}|\mathbf{X}\right\} = \mathrm{E}\left\{Z_i^r|\mathbf{X}\right\} \cdot p_{j,i} = \sum_{s=1}^r x_i^s \cdot d_{r-s} \cdot P_{\lambda_i} \cdot p_{j,i}\,. \tag{2.33}$$

2.B. Mean Squared Error of the Least-Squares Estimator

We aim at deriving an expression for the Mean Squared Error (MSE) per transition probability $p_{j,i}$ of the least-squares estimator in (2.8), defined as $\mathrm{MSE}_{j,i} \doteq |\hat{p}_{j,i} - p_{j,i}|^2$. In order to do so, we use additional notation, that we include in Table 2.3 for convenience.

First, note that we can write the LSDA estimator in (2.8) for a single receiver profile as

$$\hat{\mathbf{p}}_j = (\hat{\mathbf{Z}}^T\hat{\mathbf{Z}})^{-1}\hat{\mathbf{Z}}^T(\mathbf{y}_j - \delta_j^{\texttt{MIX}} \cdot \mathbf{1}_\rho)\,. \tag{2.34}$$

We first show that this estimator is unbiased. From (2.5), we know that $\mathrm{E}\{\mathbf{y}_j|\,\mathbf{X}\} = \hat{\mathbf{Z}}\cdot\mathbf{p}_j + \delta_j^{\texttt{MIX}}\cdot\mathbf{1}_N$. Now, using this together with the law of total expectation,

$$\mathrm{E}\{\hat{\mathbf{p}}_j\} = \mathrm{E}\{\mathrm{E}\{\hat{\mathbf{p}}_j|\mathbf{X}\}\} = \mathrm{E}\left\{(\hat{\mathbf{Z}}^T\hat{\mathbf{Z}})^{-1}\hat{\mathbf{Z}}^T\left(\mathrm{E}\{\mathbf{y}_j|\mathbf{X}\} - \delta_j^{\texttt{MIX}}\mathbf{1}_N\right)\right\} = \mathrm{E}\{\mathbf{p}_j\} = \mathbf{p}_j\,. \tag{2.35}$$

We define the error vector $\mathbf{e}_j \doteq \hat{\mathbf{p}}_j - \mathbf{p}_j$, and note that $\mathrm{MSE}_{j,i}$ is the i-th diagonal element of $\mathbf{C}_{e_j} \doteq \mathrm{E}\left\{\mathbf{e}_j\mathbf{e}_j^T\right\}$. Using the fact that $\mathbf{p}_j = \mathrm{E}\{\hat{\mathbf{p}}_j|\mathbf{X}\}$, as we can see from (2.35), we can expand the error vector as

$$\mathbf{e}_j \doteq \hat{\mathbf{p}}_j - \mathbf{p}_j = (\hat{\mathbf{Z}}^T\hat{\mathbf{Z}})^{-1}\hat{\mathbf{Z}}^T\left(\mathbf{y}_j - \mathrm{E}\{\mathbf{y}_j|\mathbf{X}\}\right)\,. \tag{2.36}$$

Then, we can write $\mathbf{C}_{e_j}$ as

$$\begin{aligned}\mathbf{C}_{e_j} &= \mathrm{E}\left\{(\hat{\mathbf{Z}}^T\hat{\mathbf{Z}})^{-1}\hat{\mathbf{Z}}^T\boldsymbol{\Sigma}_{\mathbf{y}_j|\mathbf{X}}\hat{\mathbf{Z}}(\hat{\mathbf{Z}}^T\hat{\mathbf{Z}})^{-1}\right\}\\ &= \mathbf{P}_\lambda^{-1}\mathrm{E}\left\{(\mathbf{X}^T\mathbf{D}^T\mathbf{D}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{D}^T\boldsymbol{\Sigma}_{\mathbf{y}_j|\mathbf{X}}\mathbf{D}\mathbf{X}(\mathbf{X}^T\mathbf{D}^T\mathbf{D}\mathbf{X})^{-1}\right\}\mathbf{P}_\lambda^{-1}\,,\end{aligned} \tag{2.37}$$

where the covariance matrix of $\mathbf{y}_j$ given $\mathbf{X}$ is

$$\boldsymbol{\Sigma}_{\mathbf{y}_j|\mathbf{X}} \doteq \mathrm{E}\left\{(\mathbf{y}_j - \mathrm{E}\{\mathbf{y}_j|\mathbf{X}\})(\mathbf{y}_j - \mathrm{E}\{\mathbf{y}_j|\mathbf{X}\})^T \mid \mathbf{X}\right\}\,. \tag{2.38}$$

In order to develop (2.37), we need to assume that $\rho\to\infty$ and use the Law of Large Numbers to make $(\mathbf{X}^T\mathbf{D}^T\mathbf{D}\mathbf{X})$ approximately independent from the observed inputs $\mathbf{X}$. This is, given that the input process X_i^r is stationary and memoryless, we can write

$$\lim_{\rho\to\infty}(\mathbf{X}^T\mathbf{D}^T\mathbf{D}\mathbf{X})/\rho \to \mathbf{R}_{xx}\,, \tag{2.39}$$

where the (m,n)-th element of $\mathbf{R}_{xx}$ is

$$(\mathbf{R}_{xx})_{m,n} = \frac{1}{\rho}\sum_{k=1}^{\rho}\sum_{r=1}^{k}\sum_{s=1}^{k}\mathrm{E}\{X_m^rX_n^s\}\,\alpha^2(1-\alpha)^{2k-r-s}\,. \tag{2.40}$$

We can easily find a matricial expression for $\mathbf{R}_{xx}$. First, using the hypotheses in (2.9),

$$\mathrm{E}\{X_m^rX_n^s\} = \begin{cases}(\lambda_m+\delta_m)^2+\lambda_m+\delta_m, & \text{if } m=n,\ r=s,\\ (\lambda_m+\delta_m)(\lambda_n+\delta_n), & \text{otherwise.}\end{cases} \tag{2.41}$$

Then, if we assume that $\rho \gg 1/\alpha$ and define $\alpha_q = \alpha/(2-\alpha)$, we can approximate this autocorrelation matrix by

$$\mathbf{R}_{xx} \approx (\mathbf{F}_\lambda+\mathbf{F}_\delta)[\mathbf{1}_{N\times N} + \alpha_q(\mathbf{F}_\lambda+\mathbf{F}_\delta)^{-1}](\mathbf{F}_\lambda+\mathbf{F}_\delta)\,, \tag{2.42}$$

where $\mathbf{F}_\lambda \doteq \text{diag}\{\lambda_1, \cdots, \lambda_N\}$ and $\mathbf{F}_\delta \doteq \text{diag}\{\delta_1, \cdots, \delta_N\}$. Its inverse, computed by applying the Sherman-Morrison formula, is

$$\mathbf{R}_{xx}^{-1} \approx \frac{1}{\alpha_q}\left((\mathbf{F}_\lambda + \mathbf{F}_\delta)^{-1} - \frac{1}{\alpha_q + \text{tr}(\mathbf{F}_\lambda + \mathbf{F}_\delta)}\mathbf{1}_{N\times N}\right), \tag{2.43}$$

where $\text{tr}(\cdot)$ denotes the trace operation. Going back to (2.37), our problem is to compute the i-th element of the diagonal of

$$\mathbf{C}_{e_j} \approx \frac{1}{\rho^2}\mathbf{P}_\lambda^{-1}\mathbf{R}_{xx}^{-1}\text{E}\left\{(\mathbf{DX})^T\mathbf{\Sigma}_{\mathbf{y}_j|\mathbf{X}}\mathbf{DX}\right\}\mathbf{R}_{xx}^{-1}\mathbf{P}_\lambda^{-1}. \tag{2.44}$$

We follow three steps:

1. Get a closed-form expression for $\mathbf{\Sigma}_{\mathbf{y}_j|\mathbf{X}}$.
2. Compute $\frac{1}{\rho}\text{E}\left\{(\mathbf{DX})^T\mathbf{\Sigma}_{\mathbf{y}_j|\mathbf{X}}\mathbf{DX}\right\}$.
3. Get the i-th element of the diagonal of $\mathbf{C}_{e_j}$.

Closed-form expression of $\mathbf{\Sigma}_{\mathbf{y}_j|\mathbf{X}}$. Since the variables $Y_{\lambda,j}^r$ and $Y_{\delta,j}^r$ are independent, we can split the computation of $\mathbf{\Sigma}_{\mathbf{y}_j|\mathbf{X}}$ into two subproblems:

1. Using the law of total variance, it can be shown that

$$\begin{aligned}\text{Var}\left\{Y_{\lambda,j}^r|\mathbf{X}\right\} &= \sum_{u=1}^{r}\sum_{i=1}^{N} x_i^u\left(P_{\lambda_i}p_{j,i}\alpha(1-\alpha)^{r-u} - P_{\lambda_i}^2p_{j,i}^2\alpha^2(1-\alpha)^{2(r-u)}\right),\\ \text{Cov}\left\{Y_{\lambda,j}^r, Y_{\lambda,j}^s|\mathbf{X}\right\} &= -\alpha^2(1-\alpha)^{r-s}\sum_{u=1}^{s}\left((1-\alpha)^{2(s-u)}\sum_{i=1}^{N} x_i^u P_{\lambda_i}^2p_{j,i}^2\right) \quad r \geq s\,.\end{aligned} \tag{2.45}$$

2. On the other hand, since the variables $Y_{\delta,j}^r$ and $Y_{\delta,j}^s$ are independent when $r \neq s$, we get

$$\begin{aligned}\text{Var}\left\{Y_{\delta,j}^r|\mathbf{X}\right\} &= \delta_j^{\texttt{MIX}},\\ \text{Cov}\left\{Y_{\delta,j}^r, Y_{\delta,j}^s|\mathbf{X}\right\} &= 0\,.\end{aligned} \tag{2.46}$$

We can therefore write $\mathbf{\Sigma}_{\mathbf{y}_j|\mathbf{X}}$ in matricial form as:

$$\mathbf{\Sigma}_{\mathbf{y}_j|\mathbf{X}} = \text{diag}\{\mathbf{DXP}_\lambda\mathbf{P}_j\mathbf{1}_N\} - \mathbf{D}\cdot\text{diag}\{\mathbf{XP}_\lambda^2\mathbf{P}_j^2\mathbf{1}_N\}\cdot\mathbf{D}^T + \delta_j^{\texttt{MIX}}\mathbf{I}_\rho\,, \tag{2.47}$$

where $\mathbf{P}_j \doteq \text{diag}\{p_{j,1}, p_{j,2}, \cdots, p_{j,N}\}$.

Computation of $\frac{1}{\rho}\mathrm{E}\left\{(\mathbf{DX})^T\mathbf{\Sigma}_{\mathbf{y}_j|\mathbf{X}}\mathbf{DX}\right\}$. Using (2.47), we can obtain $\frac{1}{\rho}\mathrm{E}\left\{(\mathbf{DX})^T\mathbf{\Sigma}_{\mathbf{y}_j|\mathbf{X}}\mathbf{DX}\right\}$ by performing matrix multiplications. We omit the full description of these steps for practicality issues and indicate that the result is:

$$\begin{aligned}&\tfrac{1}{\rho}\mathrm{E}\left\{(\mathbf{DX})^T\mathbf{\Sigma}_{\mathbf{y}_j|\mathbf{X}}\mathbf{DX}\right\} \approx \\ &(\mathbf{F}_\lambda+\mathbf{F}_\delta)\left\{(\lambda'_j-\lambda''_j+\delta_j^{\texttt{MIX}})\mathbf{1}_{N\times N}+\alpha_q\left(\mathbf{1}_{N\times N}(\mathbf{P}_j\mathbf{P}_\lambda-\mathbf{P}_j^2\mathbf{P}_\lambda^2)+(\mathbf{P}_j\mathbf{P}_\lambda-\mathbf{P}_j^2\mathbf{P}_\lambda^2)\mathbf{1}_{N\times N}\right)\right\}(\mathbf{F}_\lambda+\mathbf{F}_\delta) \\ &+(\mathbf{F}_\lambda+\mathbf{F}_\delta)\left\{\alpha_q(\lambda'_j-\lambda''_j+\delta_j^{\texttt{MIX}})\mathbf{I}_N+\alpha_s\mathbf{P}_j\mathbf{P}_\lambda-\alpha_q^2\mathbf{P}_j^2\mathbf{P}_\lambda^2-\left(\frac{\alpha_q}{\alpha_r}-1\right)\alpha_q\lambda'_j\mathbf{I}_N\right\},\end{aligned} \tag{2.48}$$

where λ'_j, λ''_j, α_q, α_r and α_s are defined in Table 2.3.

Computation of a single element in the diagonal of $\mathbf{C}_{e_j}$. The next step is plugging (2.48) and (2.43) into (2.44) and performing laborious matrix multiplications. We omit writing the whole expression that is obtained after this process and point out that the i-th element in the diagonal of $\mathbf{C}_{e_j}$, which is $\mathrm{Var}\{\hat{p}_{j,i}\}$ or, equivalently, $\mathrm{MSE}_{j,i}$, is:

$$\begin{aligned}\mathrm{MSE}_{j,i} \;&\approx \frac{1}{\rho}\cdot\frac{1}{\lambda_i}\cdot\left(1+\frac{\delta_i}{\lambda_i}\right)\cdot\left(1-\frac{\lambda_i+\delta_i}{\sum_{k=1}^N(\lambda_k+\delta_k)}\right) \\ &\cdot\left(\frac{1}{\alpha_q}\left(\sum_{k=1}^N\lambda_k p_{j,k}+\delta_j^{\texttt{MIX}}\right)-\frac{1}{\alpha_r}\sum_{k=1}^N\lambda_k P_{\lambda_k}p_{j,k}^2\right) \\ &+\frac{1}{\rho}\cdot\frac{1}{\lambda_i}\left(p_{j,i}-P_{\lambda_i}p_{j,i}^2\right),\end{aligned} \tag{2.49}$$

where we have assumed that $\lambda_i+\delta_i \ll \left(\sum_{k=1}^N(\lambda_k+\delta_k)\right)$. Finally, since we can assume $p_{j,i} \ll \sum_{k=1}^N\lambda_k p_{j,k}$, we get the expression

$$\begin{aligned}\mathrm{MSE}_{j,i} \approx \frac{1}{\rho}\cdot\frac{1}{\alpha_q}\cdot\;&\frac{1}{\lambda_i}\cdot\left(1+\frac{\delta_i}{\lambda_i}\right)\cdot\left(1-\frac{\lambda_i+\delta_i}{\sum_{k=1}^N(\lambda_k+\delta_k)}\right) \\ &\cdot\left(\sum_{k=1}^N\lambda_k p_{j,k}+\delta_j^{\texttt{MIX}}-\frac{\alpha_q}{\alpha_r}\sum_{k=1}^N\lambda_k P_{\lambda_k}p_{j,k}^2\right).\end{aligned} \tag{2.50}$$

Chapter 3

Design of Pool Mixes Against Profiling Attacks in Real Conditions

3.1. Introduction

Communication delay is one of the most important sources of anonymity in mix-based anonymous communication systems. As we explained in Section 1.1.1, mixes that delay messages between rounds, also called *pool mixes*, use this delay to break timing correlations between incoming and outgoing messages, thus making it harder for an adversary to correlate incoming and outgoing traffic. However, two mixes that use different delay strategies can provide very different protection levels, even if they both delay the messages by the same amount of time on average. It is thus important to understand how the messages inside the pool should be delayed so as to maximize the anonymity of the users.

Pool mixes can be roughly separated between those that delay each of the messages they receive independently, and those that make a decision taking all the messages into account. In this chapter, we study pool mixes that delay each message independently, and that can be characterized according to their *delay characteristic*. This delay characteristic is the function from which the random delays of the messages are drawn, and it has a big impact on the anonymity properties that the mix provides to its users.

Previous works show that, for a given distribution on the message delay (e.g., geometric distribution), higher average delays provide better protection to the

This chapter is adapted with permission from IEEE: Simon Oya, Fernando Pérez-González, and Carmela Troncoso. Design of pool mixes against profiling attacks in real conditions. IEEE/ACM Transactions on Networking*, 24(6):3662–3675, 2016.*

users [13, 22, 83]. The search for the optimal delay characteristic of the pool mix has been previously carried out in [13, 83] from an information-theoretic point of view and assuming that the user traffic follows unrealistic statistical models.

In this chapter, we adopt an estimation-theoretic approach to the analysis of pool mixes, studying how to optimize their delay characteristic so as to maximize the privacy of the users. This complements the information-theoretical approach of [13, 83] and allows us to obtain results in complex and realistic scenarios. As in the previous chapter, we are interested in understanding how to protect the users against *profiling attacks*, i.e., attacks that aim at revealing the long-term communication profiles of the users rather than finding the sender and recipient of a particular message. Our work shows that the optimal design of the delay characteristic actually depends on how users behave in the system, and therefore a user-independent solution is not optimal.

We start by presenting a novel theoretical study of mix-based systems that help us to better understand how the behavior of the users affects their privacy. We consider users with more complex and realistic behavior than in previous works, so as to find solutions that maximize the protection of users with complex/realistic behavioral traits. Based on this model, we obtain the delay function that maximizes our anonymity metric, namely the adversary's mean square error. This optimal pool mix design allows users communicating for almost three years with a global adversary eavesdropping on the communications to achieve the same level of protection as users communicating for one month through a binomial pool mix [14], one of the state-of-the-art designs. This highlights the importance of optimizing the delay characteristic in pool mixes. We validate our findings with real data, and discuss why previous theoretical analyses are not suitable in practice. The approach we follow in this chapter can be summarized in the following steps:

1. We find a theoretical model for the behavior of the users that suits real behavior.

2. We derive a formula that predicts the performance of the system in real scenarios.

3. We study which delay characteristic optimizes this formula from the defender's point of view.

4. We evaluate the designs obtained with real data and compare with the literature.

The rest of the chapter is structured as follows. In the next section, we explain the system model and notation used throughout the chapter, explain how we measure the privacy of the users and describe the real data we use to evaluate our

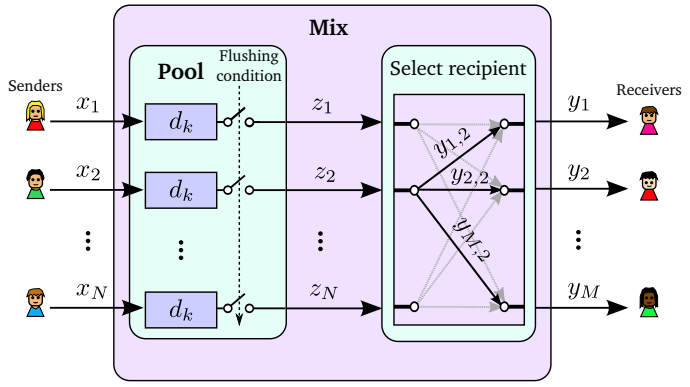


Figure 3.1: System model during the communication *round* r. A global passive adversary is only able to see the messages arriving and leaving the mix (i.e., $x_1^r, x_2^r, \cdots, x_N^r$ and $y_1^r, y_2^r, \cdots, y_M^r$) but is not aware of what happens inside of it.

findings. We propose a theoretical model for user behavior in Section 3.3, which we then use to obtain a mathematical expression that models the degree of protection of the users in the system. With this expression, we solve in Section 3.4 the problem of building an optimal delay characteristic for the pool mix and propose quasi-optimal and sub-optimal variants of this design. We evaluate our solutions and compare them with the binomial pool mix in Section 3.5, and discuss the differences between our estimation-theory approach and the information-theory approach taken in previous analysis in Section 3.6. We conclude in Section 3.7.

3.2. Preliminaries

3.2.1. System Model

Throughout this chapter, we use the same pool mix model and notation described in Chapter 2, with the simplification that all the messages sent are real (i.e., there are no dummy messages). As before, the aim of the attacker is to reconstruct the *sending profiles* of the users. For convenience, we show this simplified model in Fig. 3.1, and summarize the notation used in the chapter in Table 3.1. Please refer to Section 2.2 for a thorough description of how the mix operates and the variables that we use.

3.2.2. Privacy Metrics

As in the previous chapter, we measure the privacy of the users in our system as the Mean Squared Error (MSE) of the LSDA estimator. As we argue in

Table 3.1: Summary of notation.

Symbol	Meaning
N	Number of senders, denoted by $i \in \{1, \cdots, N\}$.
M	Number of receivers, denoted by $j \in \{1, \cdots, M\}$.
ρ	Number of rounds observed by the adversary, $r \in \{1, \cdots, \rho\}$.
$p_{j,i}$	Probability that a message from sender i is addressed to receiver j.
$\hat{p}_{j,i}$	Adversary's estimation of $p_{j,i}$.
x_i^r	Number of messages sent by sender i in round r.
z_i^r	Number of messages sent by i that leave the pool in round r.
$y_{j,i}^r$	Number of messages from i leaving to j in round r.
y_j^r	Number of messages received by j in round r.
d_k	Probability that a message is delayed k rounds in the pool.
$\mathbf{q}_i$	Sending profile of user i, $\mathbf{q}_i \doteq [p_{1,i}, p_{2,i}, \cdots, p_{M,i}]^T$.
$\mathbf{p}_j$	Vector of probabilities per receiver, $\mathbf{p}_j \doteq [p_{j,1}, \cdots, p_{j,N}]^T$.
$\mathbf{P}$	Matrix of all probabilities, $\mathbf{P} \doteq [\mathbf{p}_1, \cdots, \mathbf{p}_M]$.
$\mathbf{x}_i$	Vector with all the messages sent by i, $\mathbf{x}_i \doteq [x_i^1, \cdots, x_i^\rho]^T$.
$\mathbf{X}$	Matrix with all the input messages, $(\mathbf{X})_{r,i} \doteq x_i^r$.
$\mathbf{Z}$	Matrix with all the messages that leave the pool, $(\mathbf{Z})_{r,i} \doteq z_i^r$.
$\mathbf{y}_j$	Vector of all the messages received by j, $\mathbf{y}_j \doteq [y_j^1, y_j^2, \cdots, y_j^\rho]^T$.
$\mathbf{Y}$	Matrix with all the output messages, $(\mathbf{Y})_{r,j} \doteq y_j^r$.
$\mathbf{d}$	Delat characteristic of the mix, $\mathbf{d} \doteq [d_0, d_1, \cdots, d_{\rho-1}]^T$.
$\mathbf{D}$	Convolution matrix of the delay characteristic, shown in (2.4).
$\mathbf{E}$	Estimation error matrix, $\mathbf{E} \doteq \hat{\mathbf{P}} - \mathbf{P}$.
$\mathbf{C}_e$	Covariance matrix of the estimation error, $\mathbf{C}_e \doteq \mathrm{E}\left\{\mathbf{E}\mathbf{E}^T\right\}$.
$\mu(i)$	Avg. No of mes. sent by user i per round, $\mu(i) \doteq \mathrm{E}\left\{X_i^r\right\}$.
$\mathbf{M}$	Diagonal matrix $\mathbf{M} \doteq \mathrm{diag}\left\{[\mu(1), \cdots, \mu(N)]\right\}$.
MSE_T	Total average estimation error of the LSDA attacker.

Sect. 2.2, the reason for this is that, from all the profiling attacks, PMDA and Vida are computationally unfeasible in pool mixes. This leaves us with the attacks from the SDA family. From this family, LSDA has been proven to outperform all its relatives [34].

In the scenario without dummies, we can write (2.8) as

$$\hat{\mathbf{P}} = (\hat{\mathbf{Z}}^T\hat{\mathbf{Z}})^{-1}\hat{\mathbf{Z}}^T\mathbf{Y}\,, \tag{3.1}$$

where $\hat{\mathbf{Z}} = \mathbf{D}\cdot\mathbf{X}$.

We defined the MSE of the estimation of $p_{j,i}$ in (2.1) as $\mathrm{MSE}_{j,i} \doteq \mathrm{E}\{|\hat{p}_{j,i} - p_{j,i}|^2\}$. In this chapter, we combine the $\mathrm{MSE}_{j,i}$'s to produce a global measure of the privacy of the users in the system, that we denote by $\mathrm{MSE}_{\mathtt{T}}$. In order to produce a fair combination of the individual MSE's, we first note that the product $\rho\cdot\mu(i)\cdot\hat{p}_{j,i}$, where $\mu(i)$ is the average number of messages sent by user i per round, can be seen as an estimation of the number of messages user i sends to j during the ρ observed rounds. The MSE of this estimation can then be written as $\rho^2\mu(i)^2\mathrm{E}\{(\hat{p}_{j,i} - p_{j,i})^2\}$. Now, adding along i and j we obtain the *total* MSE of the estimated number of messages each sender sends to each receiver. Normalizing this quantity to make it comparable to the MSE of a single user profile, we obtain the *total average estimation error*:

$$\mathrm{MSE}_{\mathtt{T}} \doteq \sum_{i=1}^{N} \frac{\mu(i)^2}{\sum_{k=1}^{N}\mu(k)^2}\cdot\mathrm{MSE}_i\,. \tag{3.2}$$

where $\mathrm{MSE}_i \doteq \sum_{j=1}^{M}\mathrm{MSE}_{j,i}$. This parameter is a global metric of the level of protection of all the users against the LSDA attacker. We will use this metric to assess the performance of a pool mix with a given delay characteristic.

This metric can be expressed in a more convenient way by using the error matrix $\mathbf{E} \doteq \hat{\mathbf{P}} - \mathbf{P}$. We build the MSE matrix $\mathbf{C}_e \doteq \mathrm{E}\left\{\mathbf{E}\mathbf{E}^T\right\}$ and use the fact that the diagonal entries of this matrix correspond to MSE_i for $i = 1,\cdots,N$ to rewrite (3.2) as

$$\mathrm{MSE}_{\mathtt{T}} \doteq \mathrm{Tr}\left\{\mathbf{M}\mathbf{C}_e\mathbf{M}\right\}/\mathrm{Tr}\left\{\mathbf{M}^2\right\}\,, \tag{3.3}$$

where $\mathbf{M} \doteq \mathrm{diag}\left\{[\mu(1),\cdots,\mu(N)]\right\}$.

3.2.3. Real Datasets

In this chapter, we use real datasets to validate our theoretical study of pool mixes and to assess empirically the performance of the pool mix designs. Each dataset consists of a collection of messages exchanged in a communications system, from which we know the sending time, the sender, and the recipient. In order to work with them, we perform the following preprocessing steps:

1. We select the flushing condition of our mix, i.e., the condition that triggers the end of a round, from the two we contemplate. We consider *threshold pool mixes*, in which the end of the round is determined by the arrival of t messages to the system, and *timed pool mixes*, that wait τ units of time before triggering the end of the round. We choose values of t and τ that provide a reasonable anonymity/delay trade-off [37]: we pick $t = 100$ in the threshold pool mix in all datasets, and a value of τ in the timed pool mix that ensures that approximately 100 messages are mixed each round, but also guaranteeing that a round does not last more than 24 hours.

2. We fit our user behavioral model to the information in the datasets. The full list of parameters we use to model the sending behavior of the users and how we compute them from the datasets is explained in Section 3.3.1.

3. We simulate the mixing process as explained in the previous chapter (Section 2.2), generating the observations that would be available to the adversary: $\mathbf{X}$ and $\mathbf{Y}$.

The three datasets we use, along with the values of time τ we use for the timed mix in each case, are the following:

- **Email:** this dataset contains about 220 000 emails sent by the employees of the Enron company.[1] We treat each of the 294 email addresses sending emails as the senders of our system, and consider that messages with multiple recipients are different messages sent simultaneously to each recipient. The aim of the anonymous communication system is to hide who sends emails to whom. We use a value of $\tau = 12$ hours for the timed mix in this dataset.

- **Location:** this dataset is a collection of around 400 000 location check-ins which were carried out by the 500 most active users of the Gowalla social network.[2] Each check-in can be seen as a message sent by the sender to the location the user is checking-in, and the aim of the anonymous communication system is to hide who checks-in where. The timed mix operates with $\tau = 1$ hour.

- **MailingList:** this dataset contains almost 180 000 posts to the public mailing lists of Indymedia[3] made by the 500 most active posters. The anonymous communication system is used to hide which user posts to which thread. We use $\tau = 24$ hours.

By combining the 3 real datasets and the 2 types of flushing conditions, we get 6 sets of observations, which we use in Sects. 3.3 and 3.5.

[1] `http://www.cs.cmu.edu/~./enron/`

[2] `http://snap.stanford.edu/data/loc-gowalla.html`

[3] `http://lists.indymedia.org/`

3.3. Theoretical Study of Pool Mix-based Systems

In this section we set the theoretical grounds that we later use to improve the design of the pool mix. We start by proposing a behavioral model for the users of the mix, and then use this model to develop a formula that establishes a relation between the delay characteristic of the mix, along with the statistics of the input and output processes, and the privacy of the system.

3.3.1. Behavioral Model

We aim at proposing a statistical model that characterizes real user behavior with respect to

(a) How and when users send messages, which is determined by the random process that models the number of input messages sent by each user i in each round r, i.e., $\{X_i^r\}$.

(b) How senders choose the recipients of their messages, which is characterized by the random process that models the number of messages at each output j in round r given all the inputs, i.e., $\{Y_j^r|\mathbf{X}\}$.

3.3.1.1. Input process

For the first of these problems, we assume that the input processes $\{X_i^r\}$ for $i = 1, \ldots, N$ are stationary and ergodic, i.e., their statistical moments do not change with the rounds r, and we can compute these moments from a sufficiently large realization of the process. We do not assume that the input processes follow any specific probability distribution, which allows us to obtain distribution-independent results. We assume stationarity and ergodicity in order to be able to carry out our theoretical analysis afterwards. Nevertheless, as we will see in the next section, it is enough to assume that these properties hold up to fourth order moments since these are the moments we handle. We note that, although these assumptions limit the applicability of our results, we are able to obtain accurate results for the real data we use in this chapter and, hence, we consider these assumptions reasonable for a range of realistic scenarios as the ones we study.

3.3.1.2. Output process given the inputs

The problem with $\{Y_j^r|\mathbf{X}\}$ is different, as we need to have expressions for $\mathrm{E}\left\{Y_j^r|\mathbf{X}\right\}$ and $\mathrm{Cov}\left\{Y_j^r, Y_j^s|\mathbf{X}\right\}$ relating the inputs and the outputs to perform

the analysis. We therefore need a model that assigns the input messages to the outputs.

We propose a model that considers that the messages sent by the users in each round belong to one of two types of conversations: sporadic conversations and dedicated ones. The messages that belong to sporadic conversations are sent to a recipient chosen independently for each message. The messages that belong to a dedicated conversation are all sent to the same recipient, and this recipient may be the same across several rounds. With this model, we accommodate different sending behaviors that were considered in the literature. The independent choice of recipient, which is an appropriate model in those communication scenarios where users contact multiple receivers at once or just hold sporadic communications with different users (e.g., *Email* dataset), has been assumed in most of the previous works [22,31–35]. On the other hand, the model that considers dedicated conversations, more appropriate in systems where users hold long conversations with a single receiver before switching to another one (e.g., *Location* and *MailingList* datasets), was only used in [37], although the authors of that work did not consider that users who focus on a certain recipient are more likely to keep sending messages to that same recipient in consecutive rounds. We now describe into detail how our model works.

Model description: there are three parameters that model the sending behavior of each user i: the sending profile $\mathbf{q}_i$, which was defined before, the *focus* γ_i and the *persistence* ϵ_i. Each round r, the number of messages each user i sends, x_i^r, is assigned independently to the dedicated conversation group $x_{i,DE}^r$ with probability γ_i, and to the sporadic conversation group $x_{i,SP}^r$ otherwise. Then, all the messages in $x_{i,DE}^r$ are assigned a single recipient: this recipient is the same as the one chosen for the messages in the previous round (i.e., $x_{i,DE}^{r-1}$) with probability ϵ_i, and a new one following $\mathbf{q}_i$ otherwise. The recipient of each of the messages in $x_{i,SP}^r$ is chosen independently and according to the sending profile $\mathbf{q}_i$. This model is depicted in Fig. 3.2. Table 3.2 summarizes the new notation introduced in this section.

The rationale behind this model is the following. The focus γ_i is a probability that allows us to model users that tend to focus in a single receiver per round (γ_i close to 1), or users that are more likely to send sporadic messages to different contacts (γ_i close to 0). Intermediate values allow us to model hybrid users. The persistence ϵ_i allows us to model how likely the user is to focus on the same receiver during consecutive rounds. This value will be closer to 1, for example, for users that tend to keep long conversations with others, while it will be close to 0 for users that keep short but dedicated conversations with their recipients. This model does not account for inter-relations between users, i.e., the fact that a user choosing a certain receiver affects the choice of other users' receivers (as opposed to users choosing their recipients independently of each other). Including this feature in the system would require many additional parameters (N^2), which

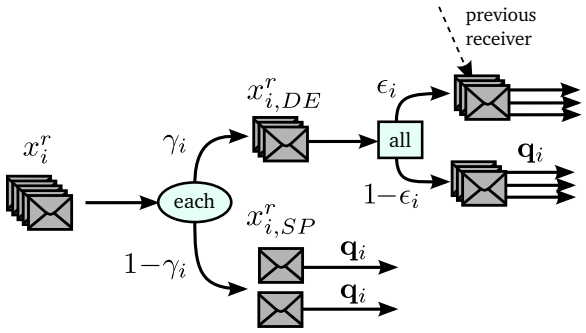


Figure 3.2: Representation of how the receivers are assigned to the messages sent by user i in round r in the proposed behavioral model.

has two disadvantages: it would substantially increase the difficulty of the privacy analysis, and obtaining these parameters given the observations would likely cause overfitting problems.

We note that, although the model does not capture scenarios where users send messages to a group of receivers (e.g., broadcast messages or dedicated conversations with multiple receivers), we obtain accurate results in presence of such traffic (e.g., results on the *Email* dataset [37]). We conjecture that these results are due to the effect of the pool, that delays messages independently and therefore group messages can be treated as sporadic messages in our analysis. In presence of more complex user sending behavior, the model should be modified by the system designer and validated following the methodology explained below.

Fitting the model to real data: We now explain how we compute the values of the parameters of our model (i.e., $\mathbf{q}_i$, γ_i and ϵ_i for all $i \in \{1, \cdots, N\}$) for each dataset and flushing condition of the mix described in Section 3.2.3. The sending profile $\mathbf{q}_i$ contains the probabilities $p_{j,i}$ that sender i sends a message to each receiver $j \in \{1, \cdots, M\}$. We compute these probabilities by counting the total number of messages user i sends to j and dividing between the total number of messages sent by user i.

Regarding the choice of γ_i and ϵ_i, we pick them so as to accurately fit the variance (and covariance) of the outputs given the inputs. First, we take into account the type of mix used and generate samples from the number of messages sent by sender i in each round r: x_i^r. Then, we store the number of messages from x_i^r that go to each receiver j in $\tilde{y}_{j,i}^r$ (note that this process is different from $y_{j,i}^r$ because it does not take the delaying in the pool into account). Let $\bar{\sigma}_i^l$ be the total sample output covariance with l rounds of difference, i.e.,

$$\bar{\sigma}_i^l \doteq \sum_{r=1}^{\rho-l} \sum_{j=1}^{M} (\tilde{y}_{j,i}^r - x_i^r \cdot p_{j,i})(\tilde{y}_{j,i}^{r+l} - x_i^{r+l} \cdot p_{j,i}) \,. \tag{3.4}$$

Table 3.2: Notation developed in Section 3.3.

Symb.	Meaning
v_i	Uniformity: $v_i \doteq 1 - \|\|\mathbf{q}_i\|\|^2$.
γ_i	Focus: prob. of sending each message to the focused receiver.
ϵ_i	Persistence: prob. of keeping the focused receiver between rounds.
$x^r_{i,DE}$	Mes. from x^r_i assigned to the dedicated conv. group.
$x^r_{i,SP}$	Mes. from x^r_i assigned to the sporadic conv. group.
σ^l_i	Total output covariance for user i with l rounds of difference.
$\bar{\sigma}^l_i$	Total sample output covariance for user i with l rounds of diff.
$r_1(i)$	Combination of v_i and γ_i; $r_1(i) \doteq (1 - v_i) + \gamma_i^2 v_i$.
$r_2(i)$	Combination of v_i and γ_i; $r_2(i) \doteq \gamma_i^2 v_i$.

Likewise, let σ^l_i be the value of the output covariance given by our model, i.e.,

$$\sigma^l_i \doteq \sum_{r=1}^{\rho-l} \sum_{j=1}^{M} \mathrm{Cov}\left\{ \tilde{Y}^r_{j,i}, \tilde{Y}^{r+l}_{j,i} | X^r_i = x^r_i, X^{r+l}_i = x^{r+l}_i \right\} . \tag{3.5}$$

This value is computed using

$$\sum_{j=1}^{M} \mathrm{Var}\left\{ \tilde{Y}^r_{j,i} | X^r_i \right\} = \left(X^r_i + X^r_i (X^r_i - 1) \gamma_i^2 \right) v_i \,, \tag{3.6}$$

and

$$\sum_{j=1}^{M} \mathrm{Cov}\left\{ \tilde{Y}^r_{j,i}, \tilde{Y}^{r+l}_{j,i} | X^r_i, X^{r+l}_i \right\} = X^r_i X^{r+l}_i \gamma_i^2 \epsilon_i^{|l|} v_i \,, \tag{3.7}$$

which are the theoretical expressions for the variance and covariance of our model, derived from the formulas (3.21) and (3.22) in the Appendix. Here, v_i represents the *uniformity* of the sending profile $\mathbf{q}_i$, and is defined as $v_i \doteq 1 - ||\mathbf{q}_i||^2$. The uniformity ranges from 0, when the profile contains one value equal to 1 and all the other values are 0, to $(N-1)/N$, when it is uniform, i.e., $p_{j,i} = 1/M \,,\ \forall j$. The first block of Table 3.2 contains a summary of the parameters that affect the variance of the outputs.

We compute γ_i for each sender i as the value that minimizes the mean squared error between the total sample variance and the variance of the model, i.e.,

$$\gamma_i = \underset{\gamma_i}{\mathrm{argmin}} \left(\bar{\sigma}^0_i - \sigma^0_i \right)^2 . \tag{3.8}$$

Similarly, we obtain the values of ϵ_i as those that minimize the error between the total sample covariance and the covariance of the model, using the γ_i obtained

in (3.8), and considering only the covariance up to R rounds of difference, i.e.,

$$\epsilon_i = \underset{\epsilon_i}{\operatorname{argmin}} \sum_{l=1}^{R} \left(\bar{\sigma}_i^l - \sigma_i^l\right)^2 . \tag{3.9}$$

We set $R = 20$ because we have validated empirically that considering more than 20 rounds of difference does not provide extra accuracy in our analysis.

Validation of the model: Figure 3.3 shows how accurate this model is: we plot the sample covariance $\mathrm{Cov}\left\{Y_{j,i}^r, Y_{j,i}^{r+l}|\mathbf{X}\right\}$ averaged over all senders i, receivers j, and rounds r, for each of the real datasets and the different mixing scenarios described in Section 3.2.3, for different values of the distance between rounds l. We also plot the average variance estimated given the inputs with the proposed model, as well as the variance predicted with the models in [37]. Note that, in the existing models in [37], it was assumed that $\mathrm{Cov}\left\{Y_{j,i}^r, Y_{j,i}^{r+l}|\mathbf{X}\right\} = 0$ for $l \neq 0$, and therefore we can only observe this value for $l = 0$ in the logarithmic plot. In all the figures, the covariance decreases as we consider rounds that are more separated. In Fig. 3.3b the covariance also oscillates. This is because the activity of the users in *Email* dataset presents a strong dependency on the time of the day (note that in this case the duration of the round is $\tau = 12$ hours, so the periodicity in the figure makes sense). The results of this figure confirm that, with the sending profile $\mathbf{q}_i$ and only two additional parameters per user (γ_i and ϵ_i), our model does not only outperform the prediction of existing models for $l = 0$, but it is also able to predict the real covariance accurately for multiple values of l.

3.3.2. Privacy Analysis

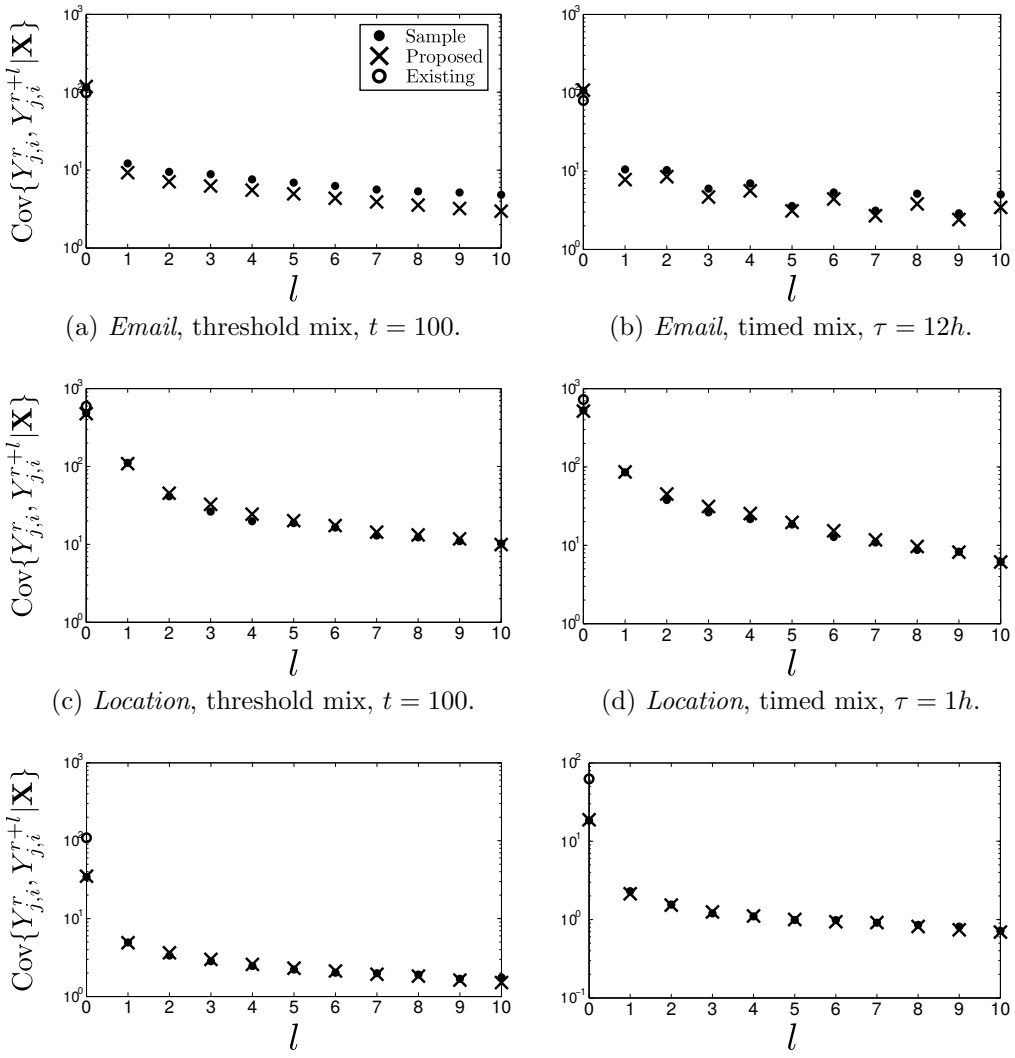
We aim at assessing the privacy of the system based on the behavioral model we have introduced. Our goal is to obtain an expression for the MSE matrix $\mathbf{C}_e$, since this can then be used to compute the total average estimation error MSE_T, defined in (3.3).

In the previous chapter, we prove that the LSDA estimator is unbiased. We briefly repeat this result here for the estimator in (3.1). In the Appendix 3.A, equation (3.24), we show that $\mathrm{E}\{\mathbf{Y}|\mathbf{X}\} = \hat{\mathbf{Z}} \cdot \mathbf{P}$, which allows us to write

$$\mathrm{E}\left\{\hat{\mathbf{P}}\right\} = \mathrm{E}\left\{(\hat{\mathbf{Z}}^T\hat{\mathbf{Z}})^{-1}\hat{\mathbf{Z}}^T\mathrm{E}\{\mathbf{Y}|\mathbf{X}\}\right\} = \mathrm{E}\left\{(\hat{\mathbf{Z}}^T\hat{\mathbf{Z}})^{-1}\hat{\mathbf{Z}}^T\hat{\mathbf{Z}} \cdot \mathbf{P}\right\} = \mathbf{P}\,.$$

Therefore, using the law of total covariance we can write $\mathbf{C}_e$ as

$$\mathbf{C}_e \doteq \mathrm{E}\left\{\mathbf{E}\mathbf{E}^T\right\} = \mathrm{E}\left\{(\hat{\mathbf{P}} - \mathbf{P})(\hat{\mathbf{P}} - \mathbf{P})^T\right\} = \mathrm{E}\left\{(\hat{\mathbf{Z}}^T\hat{\mathbf{Z}})^{-1}\hat{\mathbf{Z}}^T\boldsymbol{\Sigma}_{\mathbf{Y}|\mathbf{X}}\hat{\mathbf{Z}}(\hat{\mathbf{Z}}^T\hat{\mathbf{Z}})^{-1}\right\},$$



(a) *Email*, threshold mix, $t = 100$.

(b) *Email*, timed mix, $\tau = 12h$.

(c) *Location*, threshold mix, $t = 100$.

(d) *Location*, timed mix, $\tau = 1h$.

(e) *MailingList*, threshold mix, $t = 100$.

(f) *MailingList*, timed mix, $\tau = 24h$.

Figure 3.3: Average output covariance $\mathrm{Cov}\left\{Y^r_{j,i}, Y^{r+l}_{j,i}|\mathbf{X}\right\}$ for each of the datasets as a function of l, obtained with the real data ($\bullet$), predicted by the proposed model ($\times$), and predicted by the existing models ($\circ$). The covariance for values $l \neq 0$ in the existing models [37] is 0, and therefore it is only observable when $l = 0$.

where $\mathbf{\Sigma}_{\mathbf{Y}|\mathbf{X}}$ is a $\rho \times \rho$ matrix whose (r, s)-th entry is $\sum_{j=1}^{M} \mathrm{Cov}\left\{Y^r_j, Y^s_j|\mathbf{X}\right\}$. We now simplify the computation of $\mathbf{C}_e$ by considering that the adversary observes the system for a *sufficiently large* amount of rounds. We note that matrices $\hat{\mathbf{Z}}^T\hat{\mathbf{Z}}/\rho$ and $\hat{\mathbf{Z}}^T\mathbf{\Sigma}_{\mathbf{Y}|\mathbf{X}}\hat{\mathbf{Z}}/\rho$ contain sample averages of up to fourth order moments of the input processes. Since we are assuming that these processes are ergodic and that ρ is sufficiently large, we can approximate those matrices by their expected values. Although we could write these expected values as an expression independent from ρ, in order to reduce the notational complexity of our analysis we find it convenient to define them as $\mathbf{R}_{xx} \doteq \mathrm{E}\{\hat{\mathbf{Z}}^T\hat{\mathbf{Z}}\}/\rho$ and $\mathbf{R}_{xyx} \doteq \mathrm{E}\{\hat{\mathbf{Z}}^T\mathbf{\Sigma}_{\mathbf{Y}|\mathbf{X}}\hat{\mathbf{Z}}\}/\rho$, and

write

$$\mathbf{C}_e \approx \frac{1}{\rho}\mathbf{R}_{xx}^{-1}\mathbf{R}_{xyx}\mathbf{R}_{xx}^{-1}\,. \tag{3.10}$$

Matrix $\mathbf{R}_{xx}$ depends only on the input process ($\mathbf{X}$) and the delay characteristic (given by $\mathbf{D}$), and can be written as

$$\mathbf{R}_{xx} = \frac{1}{\rho}\mathrm{E}\left\{\mathbf{X}^T\mathbf{D}^T\mathbf{D}\mathbf{X}\right\}\,. \tag{3.11}$$

Matrix $\mathbf{R}_{xyx}$ also depends on the relations between the inputs and the outputs, represented by the covariance matrix $\mathbf{\Sigma}_{\mathbf{Y}|\mathbf{X}}$. A closed-form expression of this latter matrix can be found in (3.25) in Appendix 3.A. Plugging this formula into the definition of $\mathbf{R}_{xyx}$ above allows us to write

$$\begin{aligned}\mathbf{R}_{xyx} = &\frac{1}{\rho}\mathrm{E}\left\{\mathbf{X}^T\mathbf{D}^T\cdot\mathrm{diag}\left\{\mathbf{D}\mathbf{X}\cdot\mathbf{1}_N\right\}\cdot\mathbf{D}\mathbf{X}\right\}\\ &-\frac{1}{\rho}\mathrm{E}\left\{\mathbf{X}^T\mathbf{D}^T\mathbf{D}\cdot\mathrm{diag}\left\{\mathbf{X}\cdot\mathbf{r}_1\right\}\cdot\mathbf{D}^T\mathbf{D}\mathbf{X}\right\}\\ &+\frac{1}{\rho}\mathrm{E}\left\{\mathbf{X}^T\mathbf{D}^T\mathbf{D}\left[\sum_{i=1}^{N}\left(\mathbf{x}_i\mathbf{x}_i^T\circ\boldsymbol{E}_i\right)r_2(i)\right]\mathbf{D}^T\mathbf{D}\mathbf{X}\right\}\Bigg\}\,.\end{aligned} \tag{3.12}$$

For readability, we have grouped the effects of v_i and γ_i in the functions $r_1(i) \doteq (1-v_i)+\gamma_i^2 v_i$ and $r_2(i) \doteq \gamma_i^2 v_i$. We also use $\mathbf{r}_1 \doteq [r_1(1),\cdots,r_1(N)]^T$. For users that send messages independently to their contacts (i.e., $\gamma_i = 0$), $r_1(i) = 1-v_i$ and $r_2(i) = 0$. In contrast, users that always focus on a certain receiver (i.e., $\gamma_i = 1$) get $r_1(i) = 1$ and $r_2(i) = v_i$. Note that if $\gamma_i = 0$ for a certain user i, then $r_2(i) = 0$ and the contribution of that user to the last summand in (3.12) is zero. In that case, we can compute $\mathbf{R}_{xx}$ and $\mathbf{R}_{xyx}$ with only the first, second and third order moments of the input process of that user. However, in most scenarios this will not be the case, and we would also need the fourth order moments to compute the last summand of (3.12). Note that, although we have assumed strong stationarity and ergodicity, it is enough for our analysis to assume stationarity and ergodicity up to order four, since these are the largest order moments we handle.

We can compute our error metric $\mathrm{MSE}_{\mathtt{T}}$ to assess the privacy of the users by plugging (3.11) and (3.12) into (3.10). The complexity of this formula is considerable, and simplifying it yields much less accurate results. Fortunately, when our goal is to solve the problem of finding the delay characteristic that maximizes $\mathrm{MSE}_{\mathtt{T}}$, we can find an alternative objective function relating $\mathrm{MSE}_{\mathtt{T}}$ and $\mathbf{D}$ that is more amenable to analysis and yields a solution close to the optimal one.

3.3.3. Evaluation

We evaluate the performance of our formula in a binomial pool mix scenario [14]. The delay characteristic of this mix follows a geometric distribution $d_k = \alpha(1-\alpha)^k$, where α is the probability that a message stored in the pool leaves in each round.

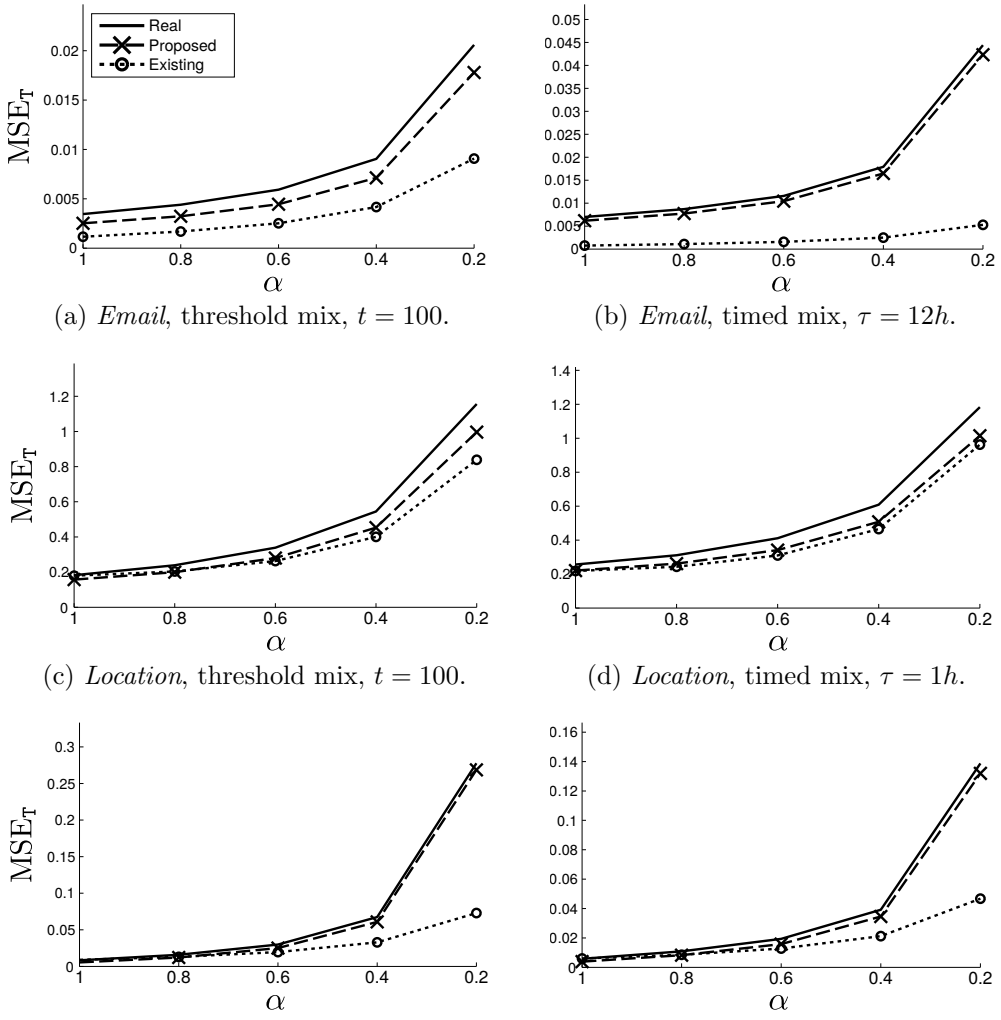
Figure 3.4 represents the overall error (3.3) predicted by our formula, together with the real error of the attack. We also plot the most accurate expressions found in the literature [37] to model the adversary's error in these datasets, which we have adapted to pool mixes. We can see that our formula clearly follows the trend of the real MSE as the delay characteristic varies, while the ones in [37] are coincidentally accurate when $\alpha = 1$ (in this case, $d_0 = 1$ and $d_k = 0$ for $k > 0$, so it is equivalent to having no pool), but are not valid to predict the error for other pool mix designs ($\alpha < 1$).

3.4. Optimizing the Design of Pool Mixes

In this section, we address the problem of optimizing the performance of the pool mix with respect to its delay characteristic, i.e., finding the delay characteristic $\mathbf{d}_{opt}$ that maximizes our global privacy metric. We start by setting an optimization problem whose solution is the optimal one, although its complexity makes it hard to study. In order to shed some light into how $\mathbf{d}_{opt}$ depends on the users' behavior, we set an alternative optimization problem which is much more amenable to analysis and whose solution is remarkably close to the optimal one. Using this alternative formulation of the problem, we study the optimal mix designs under different assumptions on the users' behavior, and come up with a user-independent albeit sub-optimal design, that is useful when no a priori information about the users is available.

3.4.1. Optimal Pool Mix Design

The optimal delay characteristic can be obtained by looking for the vector $\mathbf{d} \doteq [d_0, d_1, \cdots, d_{\rho-1}]^T$ that maximizes the overall protection of the users in the system, defined in (3.3). The problem is formally stated as



(a) *Email*, threshold mix, $t = 100$.

(b) *Email*, timed mix, $\tau = 12h$.

(c) *Location*, threshold mix, $t = 100$.

(d) *Location*, timed mix, $\tau = 1h$.

(e) *MailingList*, threshold mix, $t = 100$.

(f) *MailingList*, timed mix, $\tau = 24h$.

Figure 3.4: Overall MSE of LSDA in different realistic scenarios, as a function of the firing probability of the binomial pool mix (α), compared with the theoretical MSE predicted by our formula and the existing ones [37].

Optimal Pool Mix Design Problem:

$$\begin{aligned} \mathbf{d}_{opt} = \underset{\mathbf{d}}{\text{argmax}} \quad & \text{Tr}\left\{\mathbf{M}\mathbf{C}_e\mathbf{M}\right\} \\ \text{subject to} \quad & \sum_{k=0}^{\rho-1} d_k = 1, \quad d_k \geq 0, \ \forall k\,, \\ & \sum_{k=1}^{\rho-1} k \cdot d_k \leq \bar{\delta}\,. \end{aligned} \tag{3.13}$$

We have disregarded the normalization by $\text{Tr}\left\{\mathbf{M}^2\right\}$ in (3.3), since this normalization does not affect the maximum of the function with respect to $\mathbf{d}$. The

first constraint ensures that the delay characteristic obtained constitutes a valid probability mass function, and the second one is a constraint on the maximum average delay in rounds that the messages suffer inside the pool, where $\bar{\delta}$ denotes this maximum average delay. This formulation can also be accommodated to obtain the optimal delay function given different constraints, for example, a different bound on the maximum delay in rounds tolerated for the messages (i.e., L_{max} such that $d_k = 0$ for $k > L_{max}$).

Solving the problem in (3.13) is not straightforward: we need to know the values of a huge amount of input moments (or make assumptions on them) and all the parameters that model the sending behavior of the users, namely $\mathbf{q}_i$, γ_i and ϵ_i for $i = 1, \ldots, N$. It is also very hard to get an intuitive idea of how the shape of the optimal delay characteristic $\mathbf{d}_{opt}$ relates to these parameters. Motivated by this, in the next section we look for an alternative formulation of this problem that is more amenable to analysis.

3.4.2. Quasi-Optimal Pool Mix Design

In the technical report [84], we show that when the number of users in the system N is comparable to ρ as $\rho \to \infty$, the strategy followed to maximize $\mathrm{Tr}\{\mathbf{M}\mathbf{C}_e\mathbf{M}\}$ and $\mathrm{Tr}\{\mathbf{M}\mathbf{R}_{xx}^{-1}\mathbf{M}\}$ is the same, and therefore the delay characteristics that maximize each of these functions are similar. In that case, (3.13) can be formulated as

Quasi-optimal Pool Mix Design Problem:

$$\begin{aligned} \mathbf{d}'_{opt} = \underset{\mathbf{d}}{\operatorname{argmax}} \quad & \mathrm{Tr}\left\{\mathbf{M}\mathbf{R}_{xx}^{-1}\mathbf{M}\right\} \\ \text{subject to} \quad & \sum_{k=0}^{\rho-1} d_k = 1, \quad d_k \geq 0, \ \forall k\,, \\ & \sum_{k=1}^{\rho-1} k \cdot d_k \leq \bar{\delta}\,. \end{aligned} \tag{3.14}$$

Analyzing this problem is much easier than (3.13), as it depends on less parameters: note that we only need to consider up to second order moments of the input, and that the dependence on v_i, γ_i and ϵ_i is gone. These user parameters still affect the MSE, but they do so via terms that become independent of the delay characteristic when $N \to \infty$ is comparable to ρ. Interestingly, the solutions of (3.13) and (3.14) are very close in our real datasets, as we empirically show in Section 3.5, which indicates that we are in the case of N being comparable to ρ as $\rho \to \infty$ in all the scenarios for which we have data. We remark that for other

scenarios where $N \ll \rho$, the system designer will have to rely on (3.13) to choose the delay characteristic of the mix.

In order to provide more insight into the shape of the optimal delay characteristic when N and ρ are not comparable, we now study the solution of (3.14) under different assumptions, when $\rho \to \infty$ and $N \ll \rho$. In order to do that, we first consider that $\mathbf{R}_{xx} \approx \mathbf{\Sigma}_{xx}$ (c.f. [37]), where $\mathbf{\Sigma}_{xx}$ is the covariance matrix of the input processes $\{\hat{X}^r_{d,i}\}$, i.e., if $\mathbf{X}_c \doteq \mathbf{X} - \mathbf{1}_\rho \boldsymbol{\mu}^T$, then $\mathbf{\Sigma}_{xx} \doteq \mathrm{E}\left\{\mathbf{X}_c^T \mathbf{D}^T \mathbf{D} \mathbf{X}_c\right\}/\rho$. It will be helpful to define additional notation: the variance of the input processes is denoted by $\mu_2(i) \doteq \mathrm{Var}\{X_i^r\}$. With the variance of all users, we build $\mathbf{M}_2 \doteq \mathrm{diag}\{[\mu_2(1), \cdots, \mu_2(N)]\}$. We define the autocorrelation of the delay characteristic of the mix at lag l as $R_{dd}[l] \doteq \sum_{r=l}^{\rho-1} d_r d_{r-l}$ for $l \geq 0$, and $R_{dd}[l] = R_{dd}[-l]$ otherwise. Note that matrix $\mathbf{D}^T\mathbf{D}$ is $\rho \times \rho$ Toeplitz whose r,s-th entry is $R_{dd}[r-s]$. Based on this, we can decompose $\mathbf{\Sigma}_{xx}$ as

$$\mathbf{\Sigma}_{xx} \doteq \frac{1}{\rho}\mathrm{E}\left\{\mathbf{X}_c^T \mathbf{D}^T \mathbf{D} \mathbf{X}_c\right\} = \sum_{l=-\rho+1}^{\rho-1} \mathbf{C}_2[l] \cdot R_{dd}[l]\,, \tag{3.15}$$

where $\mathbf{C}_2[l]$ is an $N \times N$ matrix containing the covariances between all the input processes with lag l, i.e., the m,n-th entry of $\mathbf{C}_2[l]$ is $\mathrm{Cov}\left\{X_m^r, X_n^{r+l}\right\}$.

We start by assuming that the input processes are independent white processes. We then analyze how auto-correlations and cross-correlations in the input process affect the design of the optimal delay characteristic, and provide some insights into what shape this function takes when we cannot make any assumptions on the input processes.

3.4.2.1. White input processes

We start by analyzing the simple scenario where the input processes $\{X_i^r\}$ are uncorrelated and white. In that case, we have $\mathbf{C}_2[l] = \mathbf{0}_{N\times N}$ for $l \neq 0$ and $\mathbf{C}_2[0] = \mathbf{M}_2$. By using the expansion in (3.15), we get that $\mathbf{\Sigma}_{xx} = \mathbf{M}_2 \cdot R_{dd}[0]$, and therefore the optimization problem (3.14) becomes that of looking for the $\mathbf{d}$ that minimizes $R_{dd}[0]$ subject to the constraints.

This problem can be solved using the method of Lagrange multipliers. Assume that L is the largest index such that $d_k = 0$ when $k > L$. We use the fact that $d_k \geq d_{k+1}$ (otherwise, there would be another vector $\mathbf{d}$ that obtains the same value of $R_{dd}[0]$ for less average delay), and that $d_k \geq 0$ to find that $d_k = \lambda_1 - \lambda_2 \cdot k$ for $k \in \{0, \cdots, L\}$, with $\lambda_1, \lambda_2 > 0$ and $d_k = 0$ for $k > L$. This means that the values of our solution $\mathbf{d}'_{opt}$ are points of a straight line with negative slope. We then use these equations together with the constraints to find that the solution to this problem is the following:

a) Given an average delay in rounds $\bar{\delta}$, pick $L = \lceil 3\bar{\delta} \rceil$.

b) Then, set

$$d_k = \frac{2}{L+1}\left(\frac{L+1+(L-3\bar{\delta})-k}{L+2}\right), \tag{3.16}$$

for $k = 0, \cdots, L$. All the other d_k for $k > L$ are set to 0.

We refer to the pool mix implementing this delay characteristic as the *ramp* pool mix, due to the shape of the delay characteristic obtained, which we denote by $\mathbf{d}_{rmp}$. It is interesting to note that, when the inputs are white, the optimal delay function in the sense of maximizing the global MSE is user-independent as it does not depend on the input moments or the sending behavior of the users. Therefore, this design is very useful when there is no a priori information about the users.

3.4.2.2. Linear model for auto-correlations

We now assume that we can write the matrix $\mathbf{X}$ we observe as $\mathbf{X} = \mathbf{G}\tilde{\mathbf{X}}$, where $\tilde{\mathbf{X}}$ is a matrix containing uncorrelated white processes (as in the previous case) and $\mathbf{G}$ is a convolution matrix with the same structure as $\mathbf{D}$, containing in its first column the taps of the FIR filter $\mathbf{g} \doteq [g_0, g_1 \cdots, g_{\rho-1}]^T$. This filter introduces auto-correlations in the inputs processes of the users. It is straightforward to show that, in that case,

$$\mathbf{\Sigma}_{xx} = \frac{1}{\rho}\mathrm{E}\left\{\tilde{\mathbf{X}}_c^T\mathbf{G}^T\mathbf{D}^T\mathbf{D}\mathbf{G}\tilde{\mathbf{X}}_c\right\} = \mathbf{M}_2 \cdot (R_{dd}[l] * R_{gg}[l])|_{l=0}\,, \tag{3.17}$$

where $*$ denotes the convolution operation. Therefore, in this case, the optimal delay characteristic is the one that, given the constraints, minimizes $(R_{dd}[l] * R_{gg}[l])|_{l=0}$. We can compare this with the previous scenario by looking at the frequency domain. Let $\Lambda_{dd}[k]$ and $\Lambda_{gg}[k]$ be the coefficients of the ρ-point DFT of d_k and g_k, respectively. Assuming that $\mathbf{D}$ and $\mathbf{G}$ are circulant (the border effects can be disregarded as ρ grows), the optimal delay function $\mathbf{d}$ is the one that minimizes

$$(R_{dd}[l] * R_{gg}[l])|_{l=0} \approx \frac{1}{\rho}\sum_{k=0}^{\rho-1}|\Lambda_{dd}[k]|^2 \cdot |\Lambda_{gg}[k]|^2\,. \tag{3.18}$$

We could have solved the previous case (white inputs) following this frequency analysis, obtaining that the optimal delay characteristic in that case is the one that minimizes $\sum_{k=0}^{\rho-1}|\Lambda_{dd}[k]|^2$ given some delay and normalization constraints. Now, we have a specific $\Lambda_{gg}[k]$ that depends on the filter taps g_k that "colors" the input processes. The spectrum of the optimal delay characteristic $|\Lambda_{dd}[k]|^2$ will take smaller values in those frequency bins where $|\Lambda_{gg}[k]|^2$ is larger, and larger values in those bins where $|\Lambda_{gg}[k]|^2$ is smaller. In that sense, we can see the

effect of $\mathbf{g}$ as an additional constraint in the problem, that causes $\mathbf{d}$ to somehow "whiten" the input processes, while satisfying the constraints of the problem.

In this example, we have assumed that the autocorrelation of all the input processes is the same, given by the filter $\mathbf{g}$. If we have different autocorrelations per user (i.e., individual filters $\mathbf{g}(i)$ for $i = 1, \cdots, N$), formulating the problem in the same way we can see that the optimal solution consists on designing a particular delay characteristic for each user, based on the same idea above.

3.4.2.3. Linear model for cross-correlations

Similar to the previous scenario, we now assume that there is an $N \times N$ matrix $\mathbf{S}$ that generates our observation $\mathbf{X}$ by making linear combinations of N uncorrelated white processes in $\tilde{\mathbf{X}}$, i.e., $\mathbf{X} = \tilde{\mathbf{X}}\mathbf{S}$. The processes in $\mathbf{X}$ are now white and correlated processes. We assume that matrix $\mathbf{S}$ is non-singular, otherwise LSDA could not be applied directly and we would have to work in a subspace where the solution is possible. In that case,

$$\mathbf{\Sigma}_{xx} = \frac{1}{\rho}\mathbf{S}^T \cdot \mathrm{E}\left\{\tilde{\mathbf{X}}_c^T \mathbf{D}^T \mathbf{D} \tilde{\mathbf{X}}_c\right\} \cdot \mathbf{S} = \mathbf{S}^T \mathbf{M}_2 \mathbf{S} \cdot R_{dd}[0]\,, \tag{3.19}$$

and therefore the solution is again the ramp pool mix obtained in (3.16).

3.4.2.4. Generic input processes

When the observed matrix $\mathbf{X}$ cannot be written as a combination of the examples above, i.e., $\mathbf{X} = \mathbf{G}\tilde{\mathbf{X}}\mathbf{S}$, then, besides $R_{dd}[0]$, other autocorrelation terms $R_{dd}[k]$ can take part in the optimization problem. A toy example for this is the case where we have $N = 2$ white users, and user $i = 2$ always sends the same number of messages user $i = 1$ has sent in the previous round, i.e., $X_2^r = X_1^{r-1}$. This can represent, for example, a user that always replies to each message she receives in the next round, or a repeater. In this case, $\mathbf{\Sigma}_{xx} = \mathbf{I}_{2\times 2} \cdot R_{dd}[0] \cdot \mu_2(1) + (\mathbf{1}_{2\times 2} - \mathbf{I}_{2\times 2}) \cdot R_{dd}[1] \cdot \mu_2(1)$, and we obtain that the optimal delay function is the one that minimizes $R_{dd}[0] - R_{dd}[1]$ subject to the constraints. This results in a bell-shaped delay characteristic, which is far from the straight line we obtain for the cases 1 and 3 studied before.

For a generic input process, we cannot find a closed-form solution for the delay characteristic. We can only expect to find a delay function more similar to a straight line when the input correlations are small, and a bell-shaped function when the correlations between the processes are large, or even when they are small but the number of users is large.

3.5. Evaluation

In this section, we evaluate the performance of the different delay characteristics proposed in the previous sections. We build the following pool mixes, that differ on their delay characteristic:

1. The optimal pool mix, whose delay characteristic is given by the solution to (3.13), i.e., $\mathbf{d}_{opt}$.
2. The quasi-optimal pool mix, whose delay characteristic is given by the solution to (3.14) when no assumptions on the input processes are made, i.e., $\mathbf{d}'_{opt}$.
3. The ramp pool mix, whose delay characteristic, given by (3.16) and denoted by $\mathbf{d}_{rmp}$, is the solution to (3.14) under the assumptions that the input processes are white and uncorrelated.
4. The binomial pool mix, which has been widely used in the literature and claimed as the optimal pool mix in terms of anonymity in previous works [13,83]. The delay characteristic of this pool is denoted by $\mathbf{d}_{bin}$ and is given by $d_k = \alpha(1-\alpha)^k$, where α is a parameter between 0 and 1 controlling the delay of the messages inside the pool.

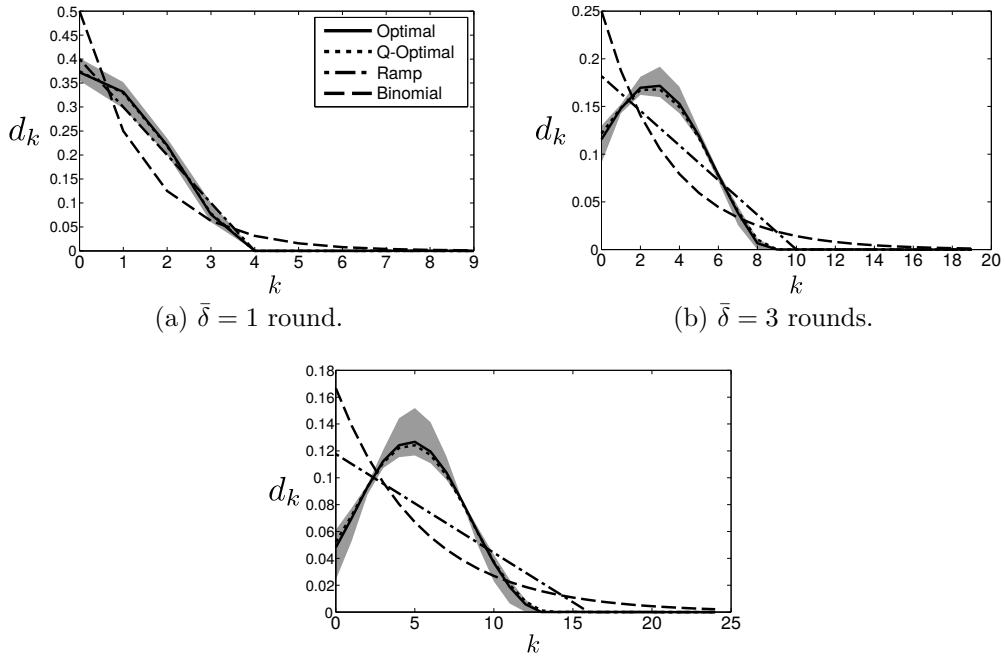
Each of these designs is assigned a flushing condition and evaluated with real data, as explained in Section 3.2.3. All the simulations are performed using Matlab software, including the optimization tools to solve (3.13) and (3.14).

3.5.1. Shape of the Delay Characteristic

We first compare the shape of the delay characteristics of the four pool mix designs, for different values of the average delay in rounds $\bar{\delta}$. This is shown in Fig. 3.5. Since $\mathbf{d}_{opt}$ and $\mathbf{d}'_{opt}$ are different for each input dataset, we plot the *average* result in the figure. The gray area represents the maximum and minimum values obtained for each $d_k \in \mathbf{d}_{opt}$ in the datasets.

The figure confirms that the average shape of the delay characteristic of the optimal and quasi-optimal designs is very similar for all the values of average delay $\bar{\delta}$ we test, which confirms our intuitions in Section 3.4.2. It is also worth noticing that these delay functions are *non-decreasing* and bell-shaped:this happens because the number of users N in the real datasets we have used for evaluation is comparable to the number of rounds observed ρ, as explained in [84].

We show in Table 3.3 the variance of the delay of each design (we show the average variance over all datasets for the optimal and quasi-optimal pool



(a) $\bar{\delta} = 1$ round.

(b) $\bar{\delta} = 3$ rounds.

(c) $\bar{\delta} = 5$ rounds.

Figure 3.5: Comparison between the delay characteristic of different pool mix designs.

Table 3.3: Expected variance of the delay (in number of rounds) introduced by each type of pool for different values of average delay.

$\bar{\delta}$	1	2	3	4	5
$Var\{\mathbf{d}_{bin}\}$	2.00	6.00	12.00	20.00	30.00
$Var\{\mathbf{d}_{rmp}\}$	1.00	3.00	6.00	10.00	15.00
$Var\{\mathbf{d}'_{opt}\}$	0.91	2.31	4.08	6.05	8.20
$Var\{\mathbf{d}_{opt}\}$	0.90	2.23	3.92	5.79	7.84

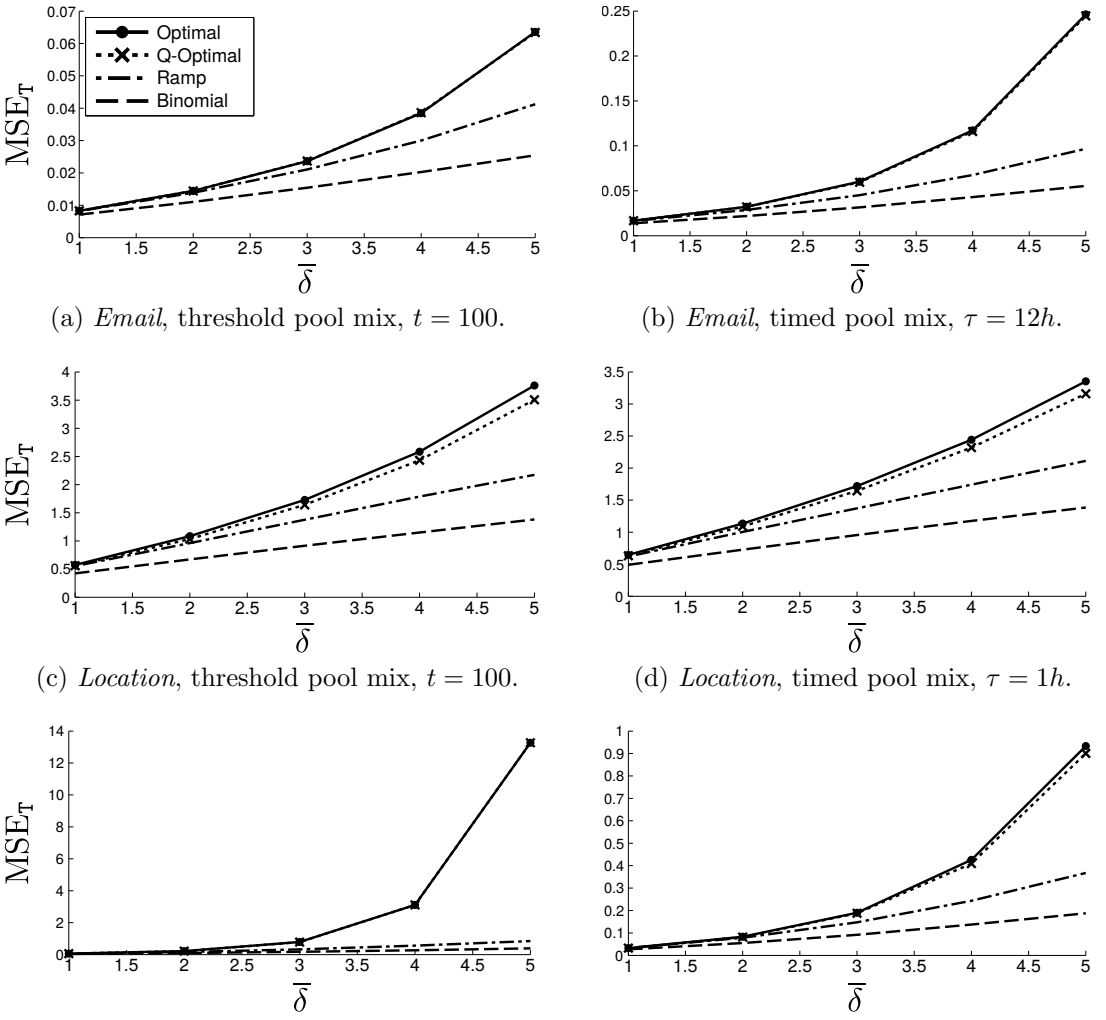
mixes). Again, the optimal and quasi-optimal designs have very similar variance, as their shape is almost the same. These pool mixes do not only maximize the error but also have the smallest variance, which means that, when using them, users can expect a delay in rounds close to the average value $\bar{\delta}$ for each of their messages, while for the other types of designs the delay is less predictable. It is also worth noticing that the variance of the ramp pool mix is half the variance of the binomial one, which makes the ramp pool mix a more appealing option when no information about the users is available to the system designer.

3.5.2. Performance of the Pool Mix Designs

We evaluate the protection that the different pool mix designs offer against the LSDA adversary. Figure 3.6 shows the global MSE (MSE_T) obtained by using the different pool mix designs for different values of average delay (we have omitted the value at $\bar{\delta} = 0$, as all the pools are equivalent in that case, i.e., $d_0 = 1$). We can see that the ramp pool mix considerably improves the protection of the users in the system when compared with the traditional binomial pool mix, but the optimal and quasi-optimal designs achieve a substantially better result. The difference between these latter is small, although the optimal pool mix performs slightly better in every case. For an average delay of $\bar{\delta} = 5$ rounds, the ratios between the MSE achieved by the optimal pool mix and the MSE achieved by the binomial pool mix for each dataset in Fig. 3.6 are, in order, 2.5, 4.4, 2.7, 2.4, 34.3 and 5.0. Since the dependence of the MSE on the number of rounds observed is $1/\rho$, we can also interpret these numbers as ratios on the number of rounds. For example, in *MailingList* dataset using a timer with $\tau = 24h$ as flushing condition and allowing a maximum average delay of $\bar{\delta} = 5$ rounds (Fig. 3.6f, ratio of 5.0), users exchanging messages during a month using a binomial pool mix would get the same degree of protection against a profiling adversary than users communicating for 5 months with our optimal pool mix. If we use a threshold of $t = 100$ as flushing condition instead, the optimal design allows users to exchange messages for almost three years while having more protection than users exchanging messages for a month with a binomial pool mix. These results highlight the importance of the delay strategy in the privacy of the system: choosing a well-designed delay characteristic can make a huge difference in the performance.

3.6. Comparison with Related Work

In this section, we compare our work with other attempts at finding the optimal delay characteristic for a pool mix. There are two works that have performed this analysis. On the one hand, Danezis analyzes in [13] the delay characteristic of a continuous pool mix [12], i.e., a pool mix that does not operate in batches or rounds, but applies to each input process $X_i(t)$ a random delay which can be modeled by a continuous probability density function $d(t)$. However, the experiments of this paper perform a time discretization, where the mix works in so-called "simulation tics". These simulation tics are equivalent to our communication rounds, so we can consider both scenarios equivalent and apply our analysis here. On the other hand, Rebollo-Monedero et al. [83] study threshold pool mixes that work by storing messages and forwarding k of them to their recipients when the pool contains $n \geq k$ of them. We have not considered this flushing condition in our cases of study, as we are considering that the flushing condition is independent of the current number of messages in the pool, but our framework can easily



(a) *Email*, threshold pool mix, $t = 100$.

(b) *Email*, timed pool mix, $\tau = 12h$.

(c) *Location*, threshold pool mix, $t = 100$.

(d) *Location*, timed pool mix, $\tau = 1h$.

(e) *MailingList*, threshold pool mix, $t = 100$.

(f) *MailingList*, timed pool mix, $\tau = 24h$.

Figure 3.6: Performance of pool mixes in different realistic scenarios and using different flushing strategies (timed and threshold mixes), as a function of the average delay ($\overline{\delta}$). Each line represents the overall MSE of LSDA ($\mathrm{MSE_T}$) using a different delay characteristic.

accommodate it.

The approach to measure anonymity used by both Danezis [13] and Rebollo-Monedero [83] is radically different from ours. They use information-theoretic metrics, mainly Shannon's entropy, to measure the anonymity of single messages; while we use an estimation-theoretic approach to measure the error of the adversary when profiling a user. The information-theoretic approach works as follows: for a target output message, it builds a probability distribution describing the likelihood that any input message corresponds with the target output. Anonymity is then measured as the entropy of this probability distribution: maximal entropy implies maximum anonymity since it represents the case where the output mes-

sage is equally likely to have come from each input; and minimal entropy (zero) indicates minimum anonymity, i.e., that the output message can unequivocally be related to an input.

Under this anonymity definition, and using Shannon's result that states that the distribution that maximizes the entropy when there is a constraint on the average delay is the geometric distribution (exponential for the continuous case), both Danezis [13] and Rebollo-Monedero et al. [83] obtain that the binomial pool mix (called exponential pool mix in the continuous case [13]) is the optimal design, i.e., the one that maximizes anonymity.

In order to arrive to this conclusion both Danezis and Rebollo-Monedero et al. make unrealistic assumptions. Danezis assumes that the arrival of messages follows a Poisson distribution, but it is known that in real scenarios this assumption is not fulfilled (e.g., see [37]). Rebollo-Monedero considers that the inter-arrival times have a common expectation and variance and they are uncorrelated. In this chapter we have shown that not only these assumptions are not met by real traffic, but also that the user auto- and cross-correlations have great impact on the adversary's error. In fact, the optimal delay function under the Shannon's entropy criterion depends on the user behavior statistics, and it is in general different for each user and/or population.

In order to show that under real traffic conditions the optimality of the binomial mix claimed in [13, 83] does not hold, we compare its performance to the ramp pool conducting the following experiment described in [13]. We consider a scenario in which there is only one sender that sends messages to one of only two possible receivers. These receivers also get messages from other users, from whom the adversary is not able to see the inputs but knows the distribution of their messages.

The attack proposed by Danezis is based on a hypothesis test: either the observed input goes to the first receiver (H_0) or to the second (H_1). In order to decide for one of the two, Danezis computes a log-likelihood ratio $\log L_{H_0/H_1}$. Given a threshold η, the adversary decides H_0 when $\log L_{H_0/H_1} > \eta$. The choice of the threshold η depends on the number of simulation tics observed by the adversary and the desired performance: a low η would increase the probability of deciding H_0 when H_0 is true (i.e., increase the true positive rate, TPR) but it would also increase the probability of incorrectly deciding H_0 when H_1 holds (i.e., increase the false positive rate, FPR).

We have implemented this attack and simulated the experiment in Matlab.[4] For each value of threshold η, we perform 10 000 repetitions of the experiment with 1 000 simulation tics and compute the TPR and FPR for both the binomial pool mix and ramp pool mix (3.16) configured for the same average delay $\bar{\delta} = 30$

[4]For a detailed description of this experiment and the parameters used, please see Section 3.2 in [13]. In order to compute the FPR, we have also simulated the H_1 scenario.

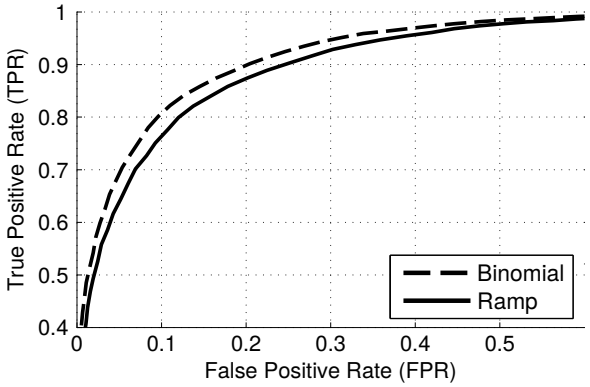


Figure 3.7: Receiver operating characteristic for Danezis's classifier in [13], given 1000 simulation tics, for the binomial pool mix and our ramp pool mix.

rounds. We plot in Figure 3.7 the receiver operating characteristic (ROC) curve, i.e., the TPR versus the FPR obtained, for both designs. We see that the ramp pool mix outperforms the binomial pool mix since, for any given TPR, the ramp pool mix always achieves a larger FPR, i.e., the adversary will wrongly choose H_0 when H_1 holds more often when the ramp pool is used.

The result of our experiments shows that, even though the binomial pool mix maximized the information-theoretic measure of sender anonymity introduced in [30], it is not optimal against the message tracing attack proposed in [13]. The reason is that information-theoretic metrics only consider the probability distribution of inputs for a given output message, disregarding the distribution of all the other messages. Hence, they do not reflect adequately how a given input blends with other incoming traffic, which is key against attacks aiming at tracing messages.

3.7. Conclusions

In this chapter, we study the design of pool mixes, the basic building blocks of high-latency anonymous communication systems. We carry out such study from an estimation-theoretic point of view, deriving a theoretical model for user behavior, which we validate with real data, and obtaining a mathematical expression for the estimation error of the best profiling adversary against pool mixes. We use this estimation error as a metric of privacy, and obtain the delay characteristic of the pool mix that maximizes this metric. Since computing this optimal design requires a lot of information, we also propose a quasi-optimal solution which is much easier to compute and to understand, although its application is more limited. Our work shows that the optimal pool mix design depends on the users' behavior, and therefore it is impossible to compute it when no information

about the users is available. In order to solve this, we also propose the ramp pool mix, a sub-optimal but user-independent design that is useful when the number of rounds observed is much larger than the number of users in the system.

We compare the performance of our proposals and the state-of-the-art binomial pool mix against a profiling adversary, and show that our constructions substantially increase the protection provided to users. We further show that, contrary to prior belief [13, 83], the binomial pool mix is neither optimal against message-tracing attacks.

Appendix

3.A. Second-Order Moments of Outputs, Given the Inputs

Our goal is to derive expressions for the expected value and the second-order moments of the outputs $Y_{j,i}^r$ given the inputs $\mathbf{X}$. To make the derivations easier, in this section we use the random variable $\tilde{Y}_{j,i}^u$, that models the number of messages sent by sender i in round u, that are addressed to receiver j (but can reach them in another round, since they may be delayed inside the pool). We also build the vector $\tilde{\mathbf{y}} \doteq [\tilde{y}_{j,i}^1, \tilde{y}_{j,i}^2, \cdots, \tilde{y}_{j,i}^\rho]^T$. Those messages enter the pool, and leave in that round or in the subsequent ones. We define $Y_{j,i}^{r,u}$ as the number of those messages leaving in round r. Note that $Y_{j,i}^r = \sum_{u=1}^r Y_{j,i}^{r,u}$. When there is no pool, we also have $Y_{j,i}^r = \tilde{Y}_{j,i}^r$. We also use $v_{j,i} \doteq p_{j,i}(1-p_{j,i})$ and note that $v_i = \sum_{j=1}^M v_{j,i}$.

We start by building the relations between $\tilde{Y}_{j,i}^u$ and the inputs. These can be easily established by looking at Fig. 3.2.

$$\begin{aligned} \mathrm{E}\left\{\tilde{Y}_{j,i}^r|\mathbf{X}\right\} &= \mathrm{E}\left\{\mathrm{E}\left\{\tilde{Y}_{j,i}^r|X_{i,SP}^r, X_{i,DE}^r\right\}|X_i^r\right\} = \mathrm{E}\left\{(X_{i,SP}^r + X_{i,DE}^r)\cdot p_{j,i}|X_i^r\right\} \\ &= \mathrm{E}\left\{X_i^r \cdot p_{j,i}|X_i^r\right\} = X_i^r \cdot p_{j,i}\,. \end{aligned} \tag{3.20}$$

Since $\mathrm{E}\left\{\tilde{Y}_{j,i}^r|X_{i,SP}^r, X_{i,DE}^r\right\} = X_i^r \cdot p_{j,i}$, then the variance of this expected value conditioned on X_i^r is zero. Therefore,

$$\begin{aligned} \mathrm{Var}\left\{\tilde{Y}_{j,i}^r|\mathbf{X}\right\} &= \mathrm{E}\left\{\mathrm{Var}\left\{\tilde{Y}_{j,i}^r|X_{i,SP}^r, X_{i,DE}^r\right\}|X_i^r\right\} \\ &= \mathrm{E}\left\{\left(X_{i,SP}^r + (X_{i,DE}^r)^2\right)\cdot v_{j,i}|X_i^r\right\} \\ &= \left(X_i^r(1-\gamma_i) + (X_i^r\gamma_i)^2 + X_i^r\gamma_i(1-\gamma_i)\right)v_{j,i} \\ &= \left(X_i^r + X_i^r(X_i^r-1)\gamma_i^2\right)v_{j,i}\,. \end{aligned} \tag{3.21}$$

Similarly, it can be shown that

$$\mathrm{Cov}\left\{\tilde{Y}_{j,i}^{r}, \tilde{Y}_{j,i}^{r+l}|\mathbf{X}\right\} = \mathrm{E}\left\{X_{i,DE}^{r} X_{i,DE}^{r+l} \epsilon_i^{|l|} v_{j,i}|X_i^r, X_i^{r+l}\right\} = X_i^r X_i^{r+l} \gamma_i^2 \epsilon_i^{|l|} v_{j,i}\,. \tag{3.22}$$

Now, we show the relations between $Y_{j,i}^r$ and $\tilde{\mathbf{y}}$ in the following equations, where we use that $Y_{j,i}^{r,u}|\tilde{\mathbf{y}}$ and $Y_{j,i}^{r+l,t}|\tilde{\mathbf{y}}$ are uncorrelated for any l when $u \neq t$:

$$\mathrm{E}\left\{Y_{j,i}^{r}|\tilde{\mathbf{y}}\right\} = \sum_{u=1}^{r} \mathrm{E}\left\{Y_{j,i}^{r,u}|\tilde{Y}_{j,i}^{u}\right\} = \sum_{u=1}^{r} \tilde{Y}_{j,i}^{u} \cdot d_{r-u}\,. \tag{3.23}$$

$$\begin{aligned}
\mathrm{Var}\left\{Y_{j,i}^{r}|\tilde{\mathbf{y}}\right\} &= \sum_{u=1}^{r}\sum_{t=1}^{r} \mathrm{Cov}\left\{Y_{j,i}^{r,u}, Y_{j,i}^{r,t}|\tilde{\mathbf{y}}\right\} \\
&= \sum_{u=1}^{r} \mathrm{Var}\left\{Y_{j,i}^{r,u}|\tilde{\mathbf{y}}\right\} \\
&= \sum_{u=1}^{r} \tilde{Y}_{j,i}^{u} \cdot d_{r-u}(1 - d_{r-u})\,. \\
\mathrm{Cov}\left\{Y_{j,i}^{r}, Y_{j,i}^{s}|\tilde{\mathbf{y}}\right\} &= \sum_{u=1}^{r}\sum_{t=1}^{s} \mathrm{Cov}\left\{Y_{j,i}^{r,u}, Y_{j,i}^{s,t}|\tilde{\mathbf{y}}\right\} \\
&= \sum_{u=1}^{\min(r,s)} \mathrm{Cov}\left\{Y_{j,i}^{r,u}, Y_{j,i}^{s,u}|\tilde{\mathbf{y}}\right\} \\
&= -\sum_{u=1}^{\min(r,s)} \tilde{Y}_{j,i}^{u} \cdot d_{r-u} d_{s-u}\,.
\end{aligned}$$

We can now get the results we were looking for. Combining equations (3.23) and (3.20), we get $\mathrm{E}\left\{Y_{j,i}^{r}|\mathbf{X}\right\} = \sum_{u=1}^{r} X_i^u d_{r-u} p_{j,i}$ or, in matricial form,

$$\mathrm{E}\left\{\mathbf{Y}|\mathbf{X}\right\} = \mathbf{D} \cdot \mathbf{X} \cdot \mathbf{P} = \hat{\mathbf{Z}} \cdot \mathbf{P}\,. \tag{3.24}$$

Likewise, using the law of total variance together with the equations above we can get closed-form expressions for $\mathrm{Var}\left\{Y_{j,i}^{r}|\mathbf{X}\right\}$ and $\mathrm{Cov}\left\{Y_{j,i}^{r}, Y_{j,i}^{s}|\mathbf{X}\right\}$. These expressions are too long and we do not need them for the purpose of this thesis, so we just note that, added along j, they can be written in matricial form as

$$\begin{aligned}
\mathbf{\Sigma}_{\mathbf{Y}|\mathbf{X}} = \mathrm{diag}\left\{\mathbf{D}\mathbf{X} \cdot \mathbf{1}_N\right\} - \mathbf{D} \cdot \mathrm{diag}\left\{\mathbf{X} \cdot \mathbf{r}_1\right\} \cdot \mathbf{D}^T \\
+ \mathbf{D} \cdot \left[\sum_{i=1}^{N} \left(\mathbf{x}_i \mathbf{x}_i^T \circ \boldsymbol{E}_i\right) \cdot r_2(i)\right] \cdot \mathbf{D}^T\,.
\end{aligned} \tag{3.25}$$

The definition of $\mathbf{r}_1$ and $r_2(i)$ can be found after (3.12) in Section 3.3.2.

Part II

Obfuscation-Based Location Privacy

Chapter 4

Revisiting Location Privacy Metrics

4.1. Introduction

As we have mentioned in the introductory chapter, Location Privacy Preserving Mechanisms (LPPMs) are designed towards providing a certain notion of location privacy. In order to optimize their designs, it is important to have metrics that quantify the amount of location privacy that LPPMs provide. There is no "correct" or universal location privacy metric, as different applications have different privacy needs or goals. In the literature, we find two main trends regarding LPPM metrics: works that use the adversary's *correctness* as privacy metric, and works that rely on the *geo-indistinguishability* privacy notion.

The adversary's correctness, i.e., how close the adversary's estimate is to the correct answer, was recommended by Shokri et al. [51] to evaluate location privacy. The adversary's correctness is measured as her expected estimation error, where this error is modeled using some distance metric between the real location and the adversary's estimation [85]. This privacy metric is arguably the most popular one, since it is intuitive, easy to evaluate, and easy to operate with. Works that adopt this metric normally consider a Bayesian modeling of the adversary, i.e., an adversary with some prior knowledge about the user's movement patterns that leverages this knowledge to estimate the user's real locations [46, 65–67].

Second, there are approaches that provide privacy guarantees independent of the adversary's prior knowledge based on *geo-indistinguishability* [45]. Geo-

This chapter is adapted with permission from ACM: Simon Oya, Carmela Troncoso, and Fernando Pérez-González. Back to the drawing board: Revisiting the design of optimal location privacy-preserving mechanisms. In Proc. of Computer and Communications Security *(CCS), pages 1959–1972. ACM, 2017.*

indindistinguishability is an adaptation of differential privacy [68] to location privacy, used by a number of works [74, 86, 87]. Geo-indindistinguishability can be achieved optimally in terms of utility using expensive linear programming [65], or suboptimally using efficient remapping techniques that increase the utility of the query [67]. Finally, the Bayesian and the geo-indistinguishability approaches have been combined by Shokri [72] to obtain LPPMs that guarantee geo-indindistinguishability while achieving a good performance against the Bayesian adversary.

In this chapter, we aim at understanding the properties of the LPPMs output by these design strategies. We find that, when the target privacy notion is the adversary's expected estimation error (i.e., Shokri's correctness), there are infinite optimal LPPMs that meet a desired quality loss constraint. While this may seem advantageous, we show that following such an optimization objective may result in the selection of naive LPPMs that obviously provide little privacy, e.g., alternating the exposure of the actual user location and a far away location. Indeed, this mechanism complies on average with the constraints of the problem. Yet, it results on little uncertainty for the adversary, effectively providing a false perception of privacy.

To counter such effect we argue that, depending on the user's preferences, the search for an optimal location privacy-preserving mechanism needs to consider more criteria than the error, contradicting the belief established by Shokri et al. [51]. As examples of complementary metrics to guide the design of protection mechanisms we propose the use of the conditional entropy and a worst-case bound for quality loss. We provide efficient methods to construct LPPMs with respect to these criteria, and demonstrate that the remapping method introduced in [67] to improve the utility of geo-indistinguishability-based methods is in fact a straightforward generic scheme to build an optimal LPPM in terms of the expected estimation error from *any* obfuscation mechanism. We evaluate the effectiveness of several LPPMs according to different privacy criteria using two real location datasets concluding that, generally, LPPMs that are optimal for one criterion do not necessarily perform well on others.

To summarize, we make the following contributions:

- We provide a theoretical characterization of optimal location privacy-preserving mechanisms in terms of the mean adversarial error. We show that, for a given average quality loss, there is more than one optimal LPPM that maximizes the average privacy. This family of LPPMs forms a convex polytope in which different LPPMs provide different privacy guarantees.
- We demonstrate the limitations of evaluating defenses solely considering the correctness of the adversary [51], and advocate for the use of complementary criteria to guide the design of location privacy-preserving mechanisms where the privacy guarantees provided are better understood.

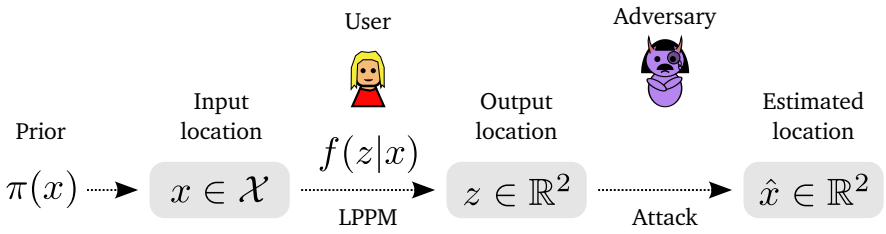


Figure 4.1: Abstraction of the location privacy problem in this chapter.

- We provide algorithms to efficiently design LPPMs based on criteria other than the adversary's error. Furthermore, we demonstrate that remapping, previously proposed as an enhancement to geo-indistinguishability, is not only beneficial to improve the utility of this technique but can be used as a generic method to turn any obfuscation LPPM into optimal in terms of average adversarial error.

- We evaluate prior and new location privacy-preserving mechanisms on two real location datasets. Our results confirm that it is difficult to find optimal LPPMs that fare well on all criteria. This demonstrates that previous approaches to design location privacy-preserving mechanisms, while having solid foundations, oversimplify the design problem and generate defenses that overestimate the level of privacy offered to the user.

This chapter is organized as follows. In Section 4.2, we introduce our system model, and the quality loss and privacy metrics we consider in this chapter. In Section 4.3 we study the consequences of choosing the average adversary error as the standard metric to evaluate location privacy, illustrating that LPPMs that are optimal by this criterion may provide little privacy. In Section 4.4 we propose to consider auxiliary metrics to avoid bad LPPM choices in the optimization. As examples, we study the use of the conditional entropy and the worst-case quality loss. We evaluate several LPPMs built according to these new criteria in Section 4.5, and offer our conclusions in Section 4.6.

4.2. System Model and Notation

We now describe our system model, which is depicted in Fig. 4.1 and is in agreement with the framework for location privacy proposed by Shokri et al. [51], and introduce the notation used throughout the chapter, which is summarized in Table 4.1.

We consider a set of users that send queries with location information to a Location Based Service (LBS) in order to obtain a service (e.g., finding points

of interest or nearby friends). As explained in Sect. 1.2.1, we consider a user-centric approach to location privacy, i.e., users run their LPPMs locally in their devices. We also assume that the users make a *sporadic* usage of the services, e.g., use applications that require infrequent location exposure. The location that a user sends in her queries can be her current location or some other location she is interested in querying about. Users wish to obtain utility from the location based service, while keeping their whereabouts private from an adversary that can observe the locations in the queries, e.g, an eavesdropper of the user-server communication, or the service provider itself. In order to protect their locations, users employ a *Location Privacy-Preserving Mechanism* (LPPM) that perturbs their location before exposing it to the server. We consider a strategic adversary that knows the LPPM, and has some knowledge about the users' movement patterns. Given the observed perturbed locations and some prior knowledge, the adversary tries to infer the users' real locations.

We model the set of locations queried by the users as a discrete set of *points of interest* denoted by $\mathcal{X} \doteq \{x_1, x_2, \cdots, x_N\}$. We refer to these locations as *real* or *input locations* since they are the actual locations that are input to the location privacy-preserving mechanism. We characterize user mobility using the *mobility profile* π. We use $\pi(x)$ to denote the *prior* probability that a user in the population queries the service provider about location x ($\pi(x) \geq 0$ and $\sum_{x \in \mathcal{X}} \pi(x) = 1$). The mobility profile can either represent the global behavior of all the users as in [67], or be tailored to a particular user, but we assume that it is known both by the user and the adversary and that it can be used to design the LPPM. We also consider independence between queries, i.e., that the input locations x from the same or other users are samples form i.i.d. random variables given by π.

The set of possible locations reported by the LPPM is denoted by $\mathcal{Z}$. We assume that users can report any location in the world $\mathcal{Z} = \mathbb{R}^2$. We refer to these locations as *output locations*, as they are the outputs of the privacy-preserving mechanism. The LPPM itself is denoted by f and modeled as a set of (continuous) conditional probability distributions, where $f(z|x)$ denotes the probability density function (pdf) of reporting the output location $z \in \mathbb{R}^2$ when the real location of the user is $x \in \mathcal{X}$ (note that $f(z|x) \geq 0$ and $\int_{\mathbb{R}^2} f(z|x)dz = 1$ for all $x \in \mathcal{X}$). We represent discrete LPPMs, i.e., LPPMs with a discrete output domain, in $\mathbb{R}^2$ with the Dirac delta function δ. For example, the LPPM that maps any $x \in \mathcal{X}$ to two particular outputs $z_1, z_2 \in \mathbb{R}^2$ with the same probability would be $f(z|x) = 0.5\delta(z - z_1) + 0.5\delta(z - z_2)$. For integration purposes, $\delta(z - z')$ must be understood as a two-dimensional Gaussian pdf centered at z' whose variance is arbitrarily small.

When using an LPPM f to obtain privacy, the user experiences a loss on the quality of service due to the fact that she reports a location that might not be the location of interest, and may even be far away from this one. We use $\mathrm{P}(f, \pi)$ to

Table 4.1: Summary of notation

Symbol	Meaning
x	Input location the user is interested in querying about.
$\mathcal{X}$	Set of valid input locations or points of interest.
z	Output location released by the LPPM, $z \in \mathbb{R}^2$.
$\hat{x}$	Adversary's estimation of the input location, $\hat{x} \in \mathbb{R}^2$.
$\pi(x)$	Prior probability that a user wants to query about x.
$f(z\|x)$	LPPM, characterized as the pdf of $z \in \mathbb{R}^2$ given $x \in \mathcal{X}$.
$f_Z(z)$	Pdf of z, i.e., $f_Z(z) = \sum_{x \in \mathcal{X}} \pi(x) \cdot f(z\|x)$.
$p(x\|z)$	Posterior probability of x given z.
$d_Q(x, z)$	Quality loss distance function between x and z.
$\overline{\mathrm{Q}}$	Average quality loss metric, in (4.1).
Q^+	Worst-case quality loss metric, in (4.2).
$d_P(x, \hat{x})$	Privacy distance function between x and $\hat{x}$.
$\mathbf{P}_{\mathtt{AE}}$	Average error privacy metric, in (4.5).
$\mathbf{P}_{\mathtt{CE}}$	Conditional entropy privacy metric, in (4.9)
$\mathbf{P}_{\mathtt{GI}}$	Geo-Indistinguishability privacy metric, in (4.11)

denote the *privacy* of the user, and $\mathrm{Q}(f, \pi)$ to denote her *quality loss*. We specify particular instantiations of these functions below.

4.2.1. Quality Loss Metrics

We consider two possible definitions of quality loss: the average loss, and the worst-case loss. To this end we introduce $d_Q(x, z)$, a function that quantifies how much quality of service is lost by a user reporting output location z when she is interested in input location x. Larger values of $d_Q(x, z)$ indicate a larger loss, and therefore a worse utility performance for the user. The canonical choice for this function is the Euclidean distance: $d_Q(x, z) = ||x - z||_2$. Note that $d_Q(\cdot)$ does not need to be a metric in the mathematical sense: it could be any function that maps an input location and a released location to a loss value (e.g., a feeling-based utility metric as in [64, 88]).

Average Loss. The average loss measures how much quality a user loses on average, and can be written as:

$$\overline{\mathrm{Q}}(f, \pi) = \sum_{x \in \mathcal{X}} \int_{\mathbb{R}^2} \pi(x) \cdot f(z|x) \cdot d_Q(x, z)\, dz\,. \tag{4.1}$$

This metric has been the typical choice of utility in the related literature [45, 46, 65–67] since it is very intuitive. This metric also has the advantage

of being linear with the LPPM f, which is very useful towards reducing the computational cost of LPPM design algorithms. Moreover, it makes the analysis of optimal algorithms in terms of average loss tractable.

Worst-case Loss. Given a function that quantifies the point-wise loss as defined above, $d_Q(x, z)$, the worst-case loss is defined as:

$$\mathrm{Q}^{+}(f, \pi) = \max_{\substack{x,z \\ \pi(x)>0 \\ f(z|x)>0}} d_Q(x, z)\,. \tag{4.2}$$

The worst-case loss measures how much utility the user loses in the worst case possible. For example, if $d_Q(x, z)$ is the Euclidean distance and the user wants to query about x, an LPPM with $\mathrm{Q}^{+}(f, \pi) \leq 2\text{km}$ ensures that the output z will not be further than 2km away from x. This property is very helpful for many applications that target nearby-type of services, since if the reported location is very far from the desired location then the result of the query would be generally useless for the user.

4.2.2. Privacy Metrics

We present now three notions of privacy: the average adversary error, the conditional entropy of the posterior distribution, and geo-indistinguishability.

Average Error. The average error is the de-facto standard to measure location privacy since Shokri et al. [51] argued that incorrectness determines the privacy of users. Consider that the adversary knows the mobility profile π and the LPPM f chosen by the user. With this information, she produces an estimate $\hat{x} \in \hat{\mathcal{X}}$ of the user's input location x. The choice of $\hat{\mathcal{X}}$ depends on the computational power of the adversary. Since we assume that the user has the freedom to report any location in $\mathbb{R}^2$, we also assume an unbounded adversary that can estimate locations on the whole world $\hat{\mathcal{X}} = \mathbb{R}^2$. Upon observing z, the adversary can build a *posterior* probability mass function over the inputs, denoted as $p(x|z)$:

$$p(x|z) = \frac{\pi(x) \cdot f(z|x)}{\sum_{x' \in \mathcal{X}} \pi(x') \cdot f(z|x')}\,. \tag{4.3}$$

Let $d_P(x, \hat{x})$ be a function that quantifies the magnitude of the adversary's error when deciding that the input location was $\hat{x}$ when the input location is actually x. As in the case of the average loss $\overline{\mathrm{Q}}$, this function $d_P(\cdot)$ does not necessarily need to be a metric (e.g., it can include the user sensitivity to an adversary learning semantic information such as in [64]). Given an output location z, the optimal

decision for the adversary in terms of minimizing the average error is

$$\hat{x}(z) = \underset{\hat{x}\in\mathbb{R}^2}{\operatorname{argmin}} \left\{ \sum_{x\in\mathcal{X}} p(x|z) \cdot d_P(x, \hat{x}) \right\} . \tag{4.4}$$

The average adversary's error, or just average error, is defined as the mean error incurred by an adversary that chooses the estimation $\hat{x}$ optimally given each observed z. Let $f_Z(z) = \sum_{x\in\mathcal{X}} \pi(x) \cdot f(z|x)$ be the probability density function of z. Then, the average error is:

$$\mathbf{P}_{\texttt{AE}}(f, \pi) = \int_{\mathbb{R}^2} f_Z(z) \sum_{x\in\mathcal{X}} p(x|z) \cdot d_P(x, \hat{x}(z))\, dz \tag{4.5}$$

$$= \int_{\mathbb{R}^2} \min_{\hat{x}\in\mathbb{R}^2} \left\{ \sum_{x\in\mathcal{X}} \pi(x) \cdot f(z|x) \cdot d_P(x, \hat{x}) \right\} dz\,. \tag{4.6}$$

Note that LPPMs designed with $\mathbf{P}_{\texttt{AE}}$ inherently protect against a strategic adversary, since the metric embeds the adversary's estimation. This metric has been used as part of the design objective in previous works [46, 51], and as a way of comparing the performance in terms of privacy of LPPMs designed with other different privacy goals in mind [45, 65–67].

Conditional Entropy. The conditional entropy is an information-theoretic metric that can be used to measure the adversary's uncertainty about the user's real location when z is released. After observing z, the adversary builds the posterior $p(x|z)$ using (4.3). The uncertainty of the adversary regarding the value of x given z can be measured as the entropy of this posterior:

$$H(x|z) \doteq -\sum_{x\in\mathcal{X}} p(x|z) \cdot \log(p(x|z))\,. \tag{4.7}$$

The conditional entropy measures the *average* entropy of the posterior after z is released. Formally,

$$\mathbf{P}_{\texttt{CE}}(f, \pi) = \int_{\mathbb{R}^2} f_Z(z) \cdot H(x|z) dz\,, \tag{4.8}$$

where $f_Z(z)$ is the probability density function of z, and $H(x|z)$ is a function of z as defined in (4.7). Alternatively, using only the mobility profile π and the LPPM f, the conditional entropy can be written as

$$\mathbf{P}_{\texttt{CE}}(f, \pi) = -\sum_{x\in\mathcal{X}} \int_{\mathbb{R}^2} \pi(x) \cdot f(z|x) \cdot \log\left(\frac{\pi(x) \cdot f(z|x)}{\sum_{x'\in\mathcal{X}} \pi(x') \cdot f(z|x')} \right) dz\,. \tag{4.9}$$

Note that this metric does not depend on the geography of the problem, i.e., on the particular values of x or z. If we use the base-two logarithm in the formula,

then $\mathbf{P}_{\mathtt{CE}}$ can be interpreted as how many bits of information the adversary needs on average to completely identify x. This metric was disregarded as a possible privacy metric in [51] due to being uncorrelated with the average error. In this chapter, we challenge such conclusion showing that considering solely the correctness of the adversary may lead to the design of LPPMs that offer low privacy. We show in Section 4.4 how using the conditional entropy as a complementary privacy metric helps to avoid choosing those undesirable LPPMs.

Geo-Indistinguishability. Geo-indistinguishability is an extension of the concept of differential privacy, originally a notion of privacy in databases, to the location privacy scenario. It was originally proposed in [45] and other works have continued the research on this line [65–67]. Formally, ϵ-geo-indistinguishability requires the following condition to be fulfilled by a location privacy-preserving mechanism f,

$$\int_A f(z|x)dz \leq e^{\epsilon \cdot d_P(x,x')} \cdot \int_A f(z|x')dz\,, \quad \forall x, x' \in \mathcal{X}\,, \forall A \subseteq \mathbb{R}^2\,. \tag{4.10}$$

This requirement ensures that given an area $A \subseteq \mathbb{R}^2$, the probability of reporting a point z in that area if the original location was x over any other location x' within some distance around x, is similar, and therefore x and x' have some degree of statistical indistinguishability. In this definition, $d_P(x, x')$ is a function that quantifies how indistinguishable x and x' are: smaller values of $d_P(x, x')$ indicate a higher indistinguishability, as the constraint becomes tighter. The privacy parameter in this definition is ϵ: larger values of ϵ indicate a looser constraint that allows $f(z|x)$ and $f(z|x')$ to be more different, and therefore x and x' become more distinguishable. Smaller values of ϵ force the probability density functions $f(z|x)$ and $f(z|x')$ to be closer, providing more privacy. Note that, if for a single input location x there is a positive probability of reporting the output in a region $A \subseteq \mathbb{R}^2$, $\int_A f(z|x)dz > 0$, then that must also be true for every other input location x'. Also, note that geo-indistinguishability is independent of the mobility profile π.

The typical choice of $d_P(x, x')$ in geo-indistinguishability is the Euclidean distance [45, 65]. Many geo-indistinguishability LPPMs rely on the fact that $d_P(x, x')$ is a metric (specifically, in the fact that it satisfies the triangular inequality $d_P(x, x') \leq d_P(x, z) + d_P(x', z)$) to prove that they meet the condition in (4.10).

Although geo-indistinguishability is generally considered a privacy guarantee and not itself a metric, we can adapt it to represent an equivalent concept to our generic metric $\mathrm{P}(f, \pi)$. Given an LPPM that provides ϵ-geo-indistinguishability, it is straightforward to see that it is also ϵ'-geo-indistinguishable if $\epsilon' > \epsilon$. Since a smaller ϵ denotes more privacy, it makes sense to define the geo-indistinguishability level provided by an LPPM f according to the smallest ϵ it

guarantees. Also, since we are defining $\mathrm{P}(f, \pi)$ as a magnitude that grows with the protection of the users, we choose to define our measure of geo-indistinguishability, $\mathbf{P}_{\mathtt{GI}}(f)$, as the inverse of the smallest ϵ guaranteed by the LPPM. Given the LPPM f, we write

$$\mathbf{P}_{\mathtt{GI}}(f) = \inf_{\substack{x,x' \in \mathcal{X} \\ z \in \mathbb{R}^2}} d_P(x, x') \cdot \left| \log \frac{f(z|x)}{f(z|x')} \right|^{-1}, \tag{4.11}$$

where we assume by convention that $\log(\frac{0}{0}) = 0$ and that $d_P(x, x') = ||x - x'||_2$ is the Euclidean distance. Larger values of $\mathbf{P}_{\mathtt{GI}}$ indicate more privacy, and the LPPM guarantees $1/\mathbf{P}_{\mathtt{GI}}$-geo-indistinguishability.

4.3. Limitations of the Expected Adversary Error Based Evaluation

The most standard way to assess the location privacy provided by two LPPMs has been the evaluation of the trade-off between their average adversary error $\mathbf{P}_{\mathtt{AE}}$ and their average loss $\overline{\mathrm{Q}}$. The use of the average error as yardstick for location privacy was proposed in [51] under the general notion of correctness, and its use as a way of comparing LPPMs was followed by many of the subsequent works [45, 46, 64–67]. The choice of distance functions $d_P(\cdot)$ and $d_Q(\cdot)$ for both the average error and the average loss in these works is mostly the Euclidean distance [45, 46, 64, 65, 67] although some of them also consider the Hamming distance [46, 51, 65] or semantic distances for privacy [64, 66].

In this section, we show the problems that stem from this established *2-dimensional* evaluation approach. We start by studying the properties of LPPMs that are optimal according to these two metrics. Then, we introduce a new LPPM that we call the *coin mechanism*, and use it as an example that brings to light the flaws of judging the privacy of an LPPM by its performance in terms of average error and average loss.

4.3.1. Study of the Established LPPM Evaluation

We start our analysis by assuming that the choice of distance functions $d_P(\cdot)$ and $d_Q(\cdot)$ is the same for simplicity, which is a typical choice in related works (e.g., both are the Euclidean distance). We denote this by $d_P(\cdot) \equiv d_Q(\cdot)$. At the end of the section, we argue what happens when this is not the case. We also introduce two definitions. First, let $\mathcal{F}_{\mathrm{Q}}$ be the set of all the LPPMs that achieve an average loss smaller or equal than Q. Formally,

$$\mathcal{F}_{\mathrm{Q}} \doteq \left\{ f \,|\, \overline{\mathrm{Q}}(f, \pi) \leq \mathrm{Q} \right\}. \tag{4.12}$$

Also, let $\mathcal{F}_{\mathrm{Q}}^{\mathtt{opt}} \subseteq \mathcal{F}_{\mathrm{Q}}$ be the set of all LPPMs $f \in \mathcal{F}_{\mathrm{Q}}$ that are optimal in terms of average adversary error, i.e.,

$$\mathcal{F}_{\mathrm{Q}}^{\mathtt{opt}} \doteq \{f \mid f \in \mathcal{F}_{\mathrm{Q}}\,,\ \mathbf{P}_{\mathtt{AE}}(f,\pi) \geq \mathbf{P}_{\mathtt{AE}}(f',\pi)\ \forall f' \in \mathcal{F}_{\mathrm{Q}}\}\ . \tag{4.13}$$

We call an LPPM inside $\mathcal{F}_{\mathrm{Q}}^{\mathtt{opt}}$ optimal, since it achieves as much privacy as possible among all the LPPMs with the same quality loss. We state the following lemma:

Lemma 4.3.1. *The set of optimal LPPMs with respect to the average privacy* $\mathbf{P}_{AE}$ *and the average loss* $\overline{Q}$ *is a convex polytope.*

Proof. Let the privacy achieved by any LPPM in $\mathcal{F}_{\mathrm{Q}}^{\mathtt{opt}}$ be $\mathbf{P}_{\mathtt{opt}}(\mathrm{Q})$. Then, we can define this set as

$$\mathcal{F}_{\mathrm{Q}}^{\mathtt{opt}} = \{f \mid \mathbf{P}_{\mathtt{AE}}(f,\pi) = \mathbf{P}_{\mathtt{opt}}(\mathrm{Q}),\quad \overline{\mathrm{Q}}(f,\pi) \leq \mathrm{Q}\}\,, \tag{4.14}$$

and since $\mathbf{P}_{\mathtt{AE}}(f,\pi)$ and $\overline{\mathrm{Q}}(f,\pi)$ are linear operations with f, (4.14) can be written as an intersection of half-spaces, which forms a convex polytope. □

Note that the proof also applies to the case where $d_P(\cdot) \not\equiv d_Q(\cdot)$ (e.g., privacy as the average Hamming error of the adversary and quality loss as the average Manhattan distance). The same outcome can be derived for the conditional entropy and geo-indistinguishability, although we leave those results out of the scope of this work.

This lemma shows that there is a *family* of optimal LPPMs that lie inside a convex polytope, instead of just a single LPPM. All of them provide the same (maximal) privacy for the same quality loss constraint so, in principle, they are equally useful. In what follows, we show why this is not the case.

We start by introducing the concept of remapping. A remapping g is a function $g : \mathbb{R}^2 \to \mathbb{R}^2$ that maps an output $z \in \mathbb{R}^2$ to another output $z' \in \mathbb{R}^2$ according to the probability density function $g(z'|z)$. It is well known that if we generate an LPPM $f' = f \circ g = \int_{\mathbb{R}^2} g(z'|z) \cdot f(z|x)dz$, then the privacy of f' in terms of average error, conditional entropy or geo-indistinguishability is not smaller than that of f. This is reasonable, as the remapping g is independent from x, and thus it does not reveal any information about it. The optimal Bayesian remapping is defined as follows:

Optimal remapping: Given an LPPM f, its optimal remapping is the one that minimizes the average loss of the composition $f' = f \circ g$, i.e., $g(z'|z) = \delta(z' - r(z))$, where

$$r(z) = \operatorname*{argmin}_{z' \in \mathbb{R}^2} \sum_{x \in \mathcal{X}} \pi(x) \cdot f(z|x) \cdot d_Q(x, z')\,. \tag{4.15}$$

This remapping assigns each location z to the location $r(z)$ in (4.15), and is used in [67] as a way of improving the utility of geo-indistinguishability LPPMs.

Now, we show that it can also be used not only to reduce the quality loss of LPPMs but to achieve optimal LPPMs in terms of average error privacy:

Theorem 4.3.2. *Let g be an optimal remapping for f, and let f' be the composition $f' = f \circ g$. If $d_P(\cdot) \equiv d_Q(\cdot)$, then f' is an optimal LPPM, i.e., $f' \in \mathcal{F}^{opt}_{\overline{\mathrm{Q}}(f',\pi)}$.*

The proof is in Appendix 4.A.

This theorem provides a straightforward way of building an optimal LPPM f' from any LPPM f. The idea is to reassign each output z of f to another symbol z' such that the average quality loss is minimized. Doing this for every output ensures that the quality loss cannot be further reduced, and since the distance function used to evaluate quality loss and privacy is the same, the best estimation the adversary can do of x is just to keep the released value. Note that the $\overline{\mathrm{Q}}(f', \pi) \leq \overline{\mathrm{Q}}(f, \pi)$. This means that, in order to find an optimal LPPM f' for a target quality loss $\overline{\mathrm{Q}}(f', \pi) = \mathrm{Q}$ using the remapping strategy, one has to adjust the loss of the LPPM f (e.g., by tuning its variance if it is a noise-based LPPM) until f' achieves the desired average loss Q.

It is straightforward to see that, if the optimal remapping for an LPPM f is just doing nothing, then it means f is optimal:

Corollary 4.3.2.1. *If the optimal remapping in* (4.15) *for an LPPM f is $g(z'|z) = \delta(z' - z)$, then f is optimal for its quality loss Q, i.e., $f \in \mathcal{F}^{opt}_{Q}$.*

This is a very convenient way of proving the optimality of an LPPM when $d_P(\cdot) \equiv d_Q(\cdot)$. Another way of seeing that such LPPM is optimal, is by realizing that with this choice of metrics, the privacy is upper bounded by the quality loss $\mathbf{P}_{\mathtt{AE}}(f, \pi) \leq \overline{\mathrm{Q}}(f, \pi)$, and the upper bound is indeed achieved when an optimal LPPM is used. We note that the fact that $\mathbf{P}_{\mathtt{AE}}(f, \pi) = \overline{\mathrm{Q}}(f, \pi)$ for optimal LPPMs is not new, as it was already mentioned in [45] about the LPPMs in [46].

4.3.2. The Coin Mechanism

We now discuss the following LPPM, which we call *the coin mechanism*, and prove that it is optimal. Let z^* be the output location that minimizes the average quality loss of an LPPM that always reports that location regardless of the input x. Formally,

$$z^* \doteq \operatorname*{argmin}_{z \in \mathbb{R}^2} \sum_{x \in \mathcal{X}} \pi(x) \cdot d_Q(x, z)\,. \tag{4.16}$$

As an example, if we measure the point-to-point loss as the mean squared error $d_Q(x, z) = ||x - z||_2^2$, then z^* will be given by the mean $z^* = \sum_{x \in \mathcal{X}} \pi(x) \cdot x$. If the loss is measured as the Euclidean distance, then z^* is the geometric median

of π. Given a generic distance function $d_Q(\cdot)$, the optimal output location z^* can be computed by solving the optimization problem in (4.16).

Let Q^* be the average quality loss achieved by an LPPM that always reports z^* regardless of the input. We construct the following LPPM, which we denote $f_{\texttt{coin}}$. First, we fix a desired quality loss $\mathrm{Q} \leq \mathrm{Q}^*$ and compute $\alpha \doteq 1 - \mathrm{Q}/\mathrm{Q}^*$. Then, we build

$$f_{\texttt{coin}}(z|x) = \alpha \cdot \delta(z - x) + (1 - \alpha) \cdot \delta(z - z^*)\,, \tag{4.17}$$

where z^* is in (4.16). This LPPM can be easily explained and implemented simulating a coin flip. We first set our desired quality loss $\mathrm{Q} \leq \mathrm{Q}^*$. Note that it would not make sense to fix Q to a value larger than Q^* since we would not achieve more privacy by doing so; an LPPM that always reports z^* and has an average loss of Q^* yields the highest privacy allowed by π. Then, we compute $\alpha = 1 - \mathrm{Q}/\mathrm{Q}^*$ and set it as the probability that our coin shows heads. Assume we are interested in querying about a location $x \in \mathcal{X}$, so we flip the coin. If the coin shows heads, then we report our desired location $z = x$. If the coin hits tails, then we report z^* regardless of the value of x. It is easy to see that the average loss of (4.17) is indeed Q, by the linearity of this metric with f.

Proposition 4.3.1. *The coin mechanism obtained for quality loss Q achieves the maximum average adversarial error possible given a constraint on the average quality loss, i.e., $f_{coin}(Q) \in \mathcal{F}_Q^{opt}$, if both are measured with the same distance function $d_P(\cdot) \equiv d_Q(\cdot)$.*

The proof is straightforward using the result in Corollary 4.3.2.1.

We now reason why, even though the coin mechanism is optimal by the standards that have been used to evaluate privacy in prior works (i.e., $\mathbf{P}_{\texttt{AE}}$ and $\overline{\mathrm{Q}}$), this LPPM is hardly desirable for any user. When the coin shows heads, the adversary observes z. If $z \neq z^*$, the adversary knows for sure that the user was interested in querying about $x = z$ and therefore the user has no privacy at all. In this case, for privacy issues, there was no point in using the LPPM. When the coin shows tails, the user is mapped far away to z^*. The adversary observes z^* and has no idea where the user is, besides the mobility profile π that was already known by her. In this case, the privacy of the user is maximal, but the quality loss is very large, since z^* is almost always very far away from the user. The quality loss is so large that the utility the user gets from this realization of the LPPM can be considered zero, so we can say that there was no point in using the LPPM in this case either. We have reached the issue we mentioned earlier: there is an LPPM, optimal by classic location privacy standards [51], that is useless both from the privacy and the quality loss point of view. This shows that there is a fundamental problem with the classic way that has been used to evaluate location privacy mechanisms.

4.3.3. The Reach of This Problem

One could think that the problem of this bi-dimensional evaluation approach lies on the fact that one cannot use the same metric to measure quality loss and privacy, e.g., the Euclidean distance. However, even with different metrics, LPPMs similar to the coin can be derived. For example, if privacy is the average mean squared error and quality loss is measured as the average Manhattan distance (i.e., the l_1 norm), a deterministic LPPM that consists on reporting the real location on most of the places and mapping to the other side of the Earth in some others is optimal, due to the fact that the MSE grows quadratically with the distance, while the l_1 (or any l_p norm) does not. In our evaluation, we show an example where an LPPM optimized for $\mathbf{P}_{\texttt{AE}}$ and $\overline{\mathrm{Q}}$ with a different pair of distance functions $d_P(\cdot) \neq d_Q(\cdot)$ suffers from the coin issue. The problem does not arise from the particular distance functions $d_P(\cdot)$ and $d_Q(\cdot)$ one uses to evaluate the average error and loss, but from the fact that these metrics are *averages*, and as such they do not restrict the minimum privacy of a single use of the LPPM or the maximum quality loss of the LPPM, they just ensure that the average is good. We believe that, while evaluating the average behavior of an LPPM is not an erroneous notion per-se, it must be handled with care to avoid undesirable results, such as the coin mechanism.

As a concluding remark, we would like to note that we have shown this problem assuming that the outputs of the LPPM and the values estimated by the adversary are points in $\mathbb{R}^2$, for notational simplicity and generality. An important fraction of previous works [46, 51, 64–66] assume a discrete model where the set of output values $\mathcal{Z}$ and estimated values $\hat{\mathcal{X}}$ are the centers of a grid over the map or points of interest such as $\mathcal{X}$. In these scenarios, one can derive a similar LPPM, where hitting tails means that the user reports the location out of the allowed ones that minimizes the average error. That LPPM can also be shown to be optimal in terms of average error and loss, although it is not a desirable LPPM for any user. For completeness, we also evaluate this scenario in our experiments. The same applies to the case where instead of having discrete input locations $\mathcal{X}$, users can report any point in $\mathbb{R}^2$ (for example, a tracking or a date finder application). The coin mechanism in (4.17) can be applied directly to this scenario, and it can be shown to be optimal (changing the summations over $\mathcal{X}$ to integrals). It is clear that using the traditional evaluation approach has flaws in all these scenarios and we must find a solution to this.

4.4. Complementary LPPM Evaluation Criteria

So far we have seen that evaluating LPPMs based solely on the average error and quality loss does not reflect whether an LPPM is actually more beneficial

than another one, due to the fact that some undesirable LPPMs are deemed optimal by this approach. In this section, we propose a solution to this evaluation procedure that consists in incorporating complementary evaluation criteria that add different perspectives to the performance of an LPPM in terms of privacy and quality loss.

We propose two metrics, that are not intended to be used as a replacement of the average error and average loss but in combination with them, adding new dimensions to the privacy vs. quality loss trade-off. The first metric we propose is the conditional entropy, a privacy metric that helps detecting inconsistent LPPMs such as the coin. The second one is the worst-case loss, a quality loss metric that provides a way of staying out of LPPMs that might yield no utility for the user at all. We comment on the implementation of LPPMs that take these metrics into consideration, and propose an LPPM that maximizes the conditional entropy while being optimal in terms of average error and quality loss. We finish the section describing other alternative privacy metrics.

4.4.1. The Conditional Entropy as a Complementary Metric

4.4.1.1. Usefulness of the Conditional Entropy

One of the problems of the coin mechanism can be seen from an information-theoretic point of view. The coin is a binary LPPM, in the sense that each input location can only be mapped to itself or to a fixed point in the map. From the adversary's perspective, this means that if the coin shows heads the adversary has no uncertainty at all about the user's input location, and if it shows tails the uncertainty is maximal. The conditional entropy can be used to detect these scenarios where the adversary has no uncertainty about x. Recalling (4.8), the conditional entropy can be written as

$$\mathbf{P}_{\mathtt{CE}}(f, \pi) = \int_{\mathbb{R}^2} f_Z(z) \cdot H(x|z) dz \,, \tag{4.18}$$

where $H(x|z) \doteq -\sum_{x \in \mathcal{X}} p(x|z) \cdot \log(p(x|z))$ is the entropy of the posterior after a location z is released. It is clear that (4.18) is an average over the entropy of all the posteriors. However, contrary to the average error, the conditional entropy is an average over functions $H(z|x)$ that are strictly concave with f. This means that in order to perform well in terms of the conditional entropy, an LPPM must spread its uncertainty among every posterior $p(x|z)$ instead of achieving maximal uncertainty with some outputs and zero uncertainty with others, as the coin does.

Another interesting property of the entropy is that it is not a geographical metric. The entropy of a posterior $H(x|z)$ does not depend on the coordinates of

the input locations or the semantic information tied to them (e.g., if the location is a hospital or a club). The entropy only depends on how evenly the posterior is distributed among the input locations. This probabilistic aspect of privacy, defined as *uncertainty* in [51], cannot be captured by other privacy notions such as correctness (e.g., the average adversary error). Due to the geographic nature of the location privacy problem, we cannot judge an LPPM based solely on its entropy. However, using it as an additional dimension of privacy gives a more complete picture of the performance of an LPPM.

We would like to point out that this notion of uncertainty provided by the entropy was disregarded as a reasonable privacy metric in [51] based on the fact that, since it is not correlated with the adversary error, it does not capture how hard is for the adversary to estimate the real input location. We claim that it is indeed the fact that the entropy is not correlated with the adversary error which gives it a special value as a *complementary* metric of privacy. The same way that semantic location privacy metrics have been proposed together with geographic metrics [64, 66] to give different perspectives on the problem, the conditional entropy is a tool that gives valuable information about the protection provided by the LPPM not captured by the average error.

We would like to make two remarks regarding the entropy. First, the conditional entropy $\mathbf{P}_{\mathtt{CE}}(f, \pi)$ must be taken into account together with the mutual information $I(X; Z)$ to get a full picture of the information-theoretic properties of the LPPM. The conditional entropy represents the average amount of uncertainty the adversary has about the real location x after observing z. A small value of conditional entropy indicates low uncertainty, and therefore we might get the impression that an LPPM with such small value provides low privacy. However, it might have been possible that the entropy of the mobility profile was already low, and therefore even if the LPPM was perfect from the privacy point of view (i.e. it did not reveal any information, $I(X; Z) = 0$), there is nothing any LPPM could have done to avoid having a low conditional entropy. We must therefore take into account the mutual information or, equivalently, the entropy of the mobility profile π, when interpreting the value given by the conditional entropy.

The second remark is that the conditional entropy must not be tailored to a particular adversary with a possibly wrong knowledge of the mobility profile π. In this chapter, we have assumed that the mobility profile π models the choice of input locations by the users, and therefore the correct way of computing the entropy is by using π in the formulas above. This entropy must be regarded as the uncertainty that a very strong passive adversary with full knowledge of the behavior of the users would have when observing z.

4.4.1.2. Implementation of LPPMs with Large Conditional Entropy

We now look for an LPPM that is optimal in terms of the average error and average loss, i.e., an LPPM in $\mathcal{F}_{\mathrm{Q}}^{\mathtt{opt}}$, that also achieves as much conditional entropy as possible. This problem is equivalent to the rate-distortion problem [89] of finding a pdf $f(z|x)$ that minimizes the mutual information between x and z subject to a quality loss constraint, which can be solved iteratively by implementing the Blahut-Arimoto algorithm. For this, we must first restrict our output to a discrete alphabet $\mathcal{Z}$ for computational reasons. The more points we assign to this alphabet and the more evenly we cover the space where we want to compute the LPPM with them, the better its performance will be. Since both the input and output domains are discrete, the LPPM is determined by the probabilities of reporting z when the user is in x, that we denote by $p(z|x)$ here for clarity. We start with an initial LPPM, for example uniform mapping $p(z|x) = 1/|\mathcal{Z}|$. Then, we perform the following steps:

1. We compute the probability mass function of each the output:
$$P_Z(z) = \sum_{x \in \mathcal{X}} \pi(x) \cdot p(z|x)\,, \qquad \forall z \in \mathcal{Z}\,. \tag{4.19}$$

2. We update the LPPM as follows:
$$p(z|x) = P_Z(z) \cdot e^{-b \cdot d_Q(x,z)}\,, \qquad \forall x \in \mathcal{X}, z \in \mathcal{Z}. \tag{4.20}$$

3. We normalize the LPPM:
$$p(z|x) = \frac{p(z|x)}{\sum_{z' \in \mathcal{Z}} p(z'|x)}\,, \qquad \forall x \in \mathcal{X}, z \in \mathcal{Z}. \tag{4.21}$$
We skip this step for the outputs z with $P_Z(z) = 0$.

4. We repeat these steps until the change in the probabilities $p(z|x)$ is below some threshold.

The value of b in the second step needs to be tuned to change the quality loss of the LPPM $\overline{\mathrm{Q}}(f, \pi)$ and cannot be pre-computed to achieve an exact value of average loss. Larger values of b yield LPPMs with less quality loss, and therefore less average error privacy and less conditional entropy. Finally, we obtain our LPPM $f(z|x)$ by applying the optimal remapping to the discrete LPPM defined in $\mathcal{X} \to \mathcal{Z}$ by the probabilities $p(z|x)$. This ensures that the resulting LPPM is optimal from the adversary error privacy point of view.

We make two remarks regarding this algorithm. The first one is about its computational cost. The operations in the three steps above are not expensive as

they only include multiplications and additions. The number of elements we need to compute in order to build $p(z|x)$ is $N \doteq |\mathcal{X}| \cdot |\mathcal{Z}|$. The first step above consists of N products and additions. In the second step $e^{-b \cdot d_Q(x,z)}$ can be precomputed as b, $\mathcal{X}$ and $\mathcal{Z}$ do not change during the algorithm, so we only have to make N multiplications, and in the third step we compute $|\mathcal{X}|$ values of $\sum_{z' \in \mathcal{Z}} p(z'|x)$ and then perform N divisions. It is clear then that the cost grows with the sizes of $\mathcal{X}$ and $\mathcal{Z}$. However, the algorithm only needs to be computed once for all the users, which can be done in the cloud, and even if the mobility profile π varies we can use a previously computed algorithm as initialization of the iteration above to get a fast update of the LPPM.

The second remark is that the LPPM produced by this algorithm also satisfies $2b$-geo-indistinguishability (the proof is in Appendix 4.B). This is a byproduct property that was not part of the reasoning behind the algorithm and it does not imply that the conditional entropy and geo-indistinguishability are related. In fact, these are *fundamentally different* notions: the former is an average metric that only considers the probabilistic (and not the geographic) aspect of the problem, while the latter is a worst-case metric that also considers the geography of the problem. Also, if we truncate the optimal conditional entropy LPPM, we obtain an LPPM that is almost optimal in terms of conditional entropy but does not provide *any* level of geo-indistinguishability.

We evaluate this LPPM and others with respect to the conditional entropy and the traditional metrics in Section 4.5.

4.4.2. The Worst-Case Quality Loss as a Complementary Metric

4.4.2.1. Usefulness of the Worst-Case Quality Loss

After analyzing the privacy problems of the coin mechanism, we now turn to the utility point of view. The great drawback of the coin mechanism from the quality loss perspective is that if the coin shows tails then the server's response to the user's query will most likely be useless due to the great quality loss incurred by reporting z^*. We can think of many applications where, if the Euclidean distance between x and z is larger than a certain value, the user gets literally nothing from the server response. For example, if we are close to a point of interest x and we want to find a nearby hospital, querying about a location z in another city will likely return a useless response from the server. In that case, we could think of generating another output and query the server again because we did not get what we were hoping for. By doing so, the privacy properties of the LPPM change, and in the case of the coin it is equivalent to always revealing our true location.

A solution to this utility issue consists in imposing a worst-case quality loss constraint on the LPPM, i.e.,

$$\mathrm{Q}^{+}(f,\pi) = \max_{\substack{x,z \\ \pi(x)>0 \\ f(z|x)>0}} d_Q(x,z) \leq \mathrm{Q}^{+}_{\mathtt{max}}\,. \tag{4.22}$$

To put it simply, we want an LPPM that releases output locations within $\mathrm{Q}^{+}_{\mathtt{max}}$ from the input location, i.e., a *bounded LPPM*. The upper bound $\mathrm{Q}^{+}_{\mathtt{max}}$ would be tuned depending on the application in question, so that a user never gets a worthless result. When used together with the average error and the average loss, the worst-case loss metric reveals those LPPMs we might want to avoid using. It is easy to see that the coin mechanism, although optimal in terms of $\mathbf{P}_{\mathtt{AE}}$ and $\overline{\mathrm{Q}}$, gives a very large value of $\mathrm{Q}^{+}(f_{\mathtt{coin}},\pi)$, which manifests its uselessness.

An interesting consequence of setting a maximum worst-case quality loss constraint when designing an LPPM is that it can simplify the computational cost of the protocol that implements or computes it. For example, take the case of the works in [46, 65], where authors assume a discrete set of output locations $\mathcal{Z}$ and propose to solve a linear program to find an optimal LPPM (in terms of average error and geo-indistinguishability, respectively). The constraint in (4.22) reduces the amount of variables that need to be computed in these programs (only a subset of $\mathcal{Z}$ are possible outputs for each input $x \in \mathcal{X}$), as well as the amount of constraints, which in turn decreases drastically the computational cost of the problem. In other implementations of LPPMs, where f is not explicitly derived but computed by adding (continuous) noise and then computing a remapping using the posterior (c.f. [67]), having a worst-case quality loss constraint reduces the amount of inputs that need to be considered when computing the posterior, effectively reducing the computational cost of the algorithm.

Finally, we would like to note that this metric exposes a basic problem with geo-indistinguishability LPPMs. As mentioned before, when using a geo-indistinguishability LPPM, if a user with input location x has non-zero probability of reporting $z \in A \subseteq \mathbb{R}^2$, then when the input location is any other $x' \in \mathcal{X}$ she must assign a non-zero probability to reporting $z \in A$. This means that for any geo-indistinguishable LPPM f, the worst-case quality loss metric $\mathrm{Q}^{+}(f,\pi)$ gives a huge value and the probability of getting a useless response from the server would be larger than zero. One could argue that, given the nature of the geo-indistinguishability guarantee, the probability of reporting a location z far from x is low and decreases exponentially with the distance between them, so we could disregard such an event from happening. However, if we really truncate the LPPM to ensure that the probability of going very far is zero, then the LPPM does not provide any geo-indistinguishability guarantee at all. It is then clear that geo-indistinguishability LPPMs are problematic from the quality loss point of view, and if a user gets zero utility from a realization of the LPPM she cannot re-use it immediately, otherwise the privacy guarantee is violated. We comment on a possible solution to this problem below.

4.4.2.2. Implementation of LPPMs with Worst-Case Quality Loss Constraint

Now we set the task of designing an LPPM that achieves a good value of worst-case quality loss or, alternatively, that ensures that the worst-case quality loss is below some bound $\mathrm{Q}^{+}(f,\pi) \leq \mathrm{Q}^{+}_{\texttt{max}}$. The straightforward approach, given an LPPM f, is to truncate the LPPM (for example, by generating samples of z until one of them ensures that $d_Q(x,z) \leq \mathrm{Q}^{+}_{\texttt{max}}$, and then releasing that z). This approach is reasonable, but one must take into account that the privacy properties of this new truncated LPPM f' are not the same as the original LPPM f, and therefore they must be re-evaluated.

Another issue that concerns the design of bounded LPPMs is that a deterministic remapping (4.15) might violate a Q^{+} constraint (i.e., even if f guarantees the Q^{+} constraint, a composition $f' = f \circ g$ might not guarantee it). Finding a bounded LPPM that achieves as much privacy as an unbounded one in $\mathcal{F}^{\texttt{opt}}_{\mathrm{Q}}$ can be an impossible task, due to the fact that the polytope defined by $\mathrm{Q}^{+}(f,\pi) \leq \mathrm{Q}^{+}_{\texttt{max}}$ might be disjoint with $\mathcal{F}^{\texttt{opt}}_{\mathrm{Q}}$. However, we can lose some privacy with respect to an optimal unbounded LPPM in exchange for a better worst-case quality loss guarantee by enforcing the bounding constraint $\mathrm{Q}^{+}(f,\pi) \leq \mathrm{Q}^{+}_{\texttt{max}}$.

4.4.3. Other Complementary Metrics

Now, we finally outline other metrics that can be used together with the average error and average quality loss to assess the privacy of LPPMs, and leave the development of LPPMs taking them into account as subject for future work.

Geo-indistinguishability (4.10) inherently ensures that an input location x is mapped to a nearby location with more probability than to a far location, which solves the privacy issue we illustrated with the coin mechanism. However, this privacy notion is not compatible with a worst-case quality loss constraint by definition, due to the fact that $f(z|x) > 0$ implies $f(z|x') > 0$, $\forall x' \in \mathcal{X}$. A possible approach to solve this utility issue of geo-indistinguishability can be to relax its definition, allowing a small tolerance value $\Delta \ll 1$, i.e.,

$$\int_A f(z|x)dz \leq e^{\epsilon \cdot d_P(x,x')} \cdot \int_A f(z|x')dz + \Delta\,, \qquad \begin{matrix} \forall x, x' \in \mathcal{X}\,, \\ \forall A \subseteq \mathbb{R}^2\,. \end{matrix} \tag{4.23}$$

Other interesting metrics to assess the privacy of LPPMs are those based on the worst-case output. For example, the worst-case output average error, defined as

$$\mathbf{P}_{\texttt{WC-AE}}(f,\pi) = \min_{\substack{z \in \mathbb{R}^2 \\ f_Z(z)>0}} \min_{\hat{x} \in \mathbb{R}^2} \left\{ \sum_{x \in \mathcal{X}} \pi(x) \cdot f(z|x) \cdot d_P(x,\hat{x}) \right\}, \tag{4.24}$$

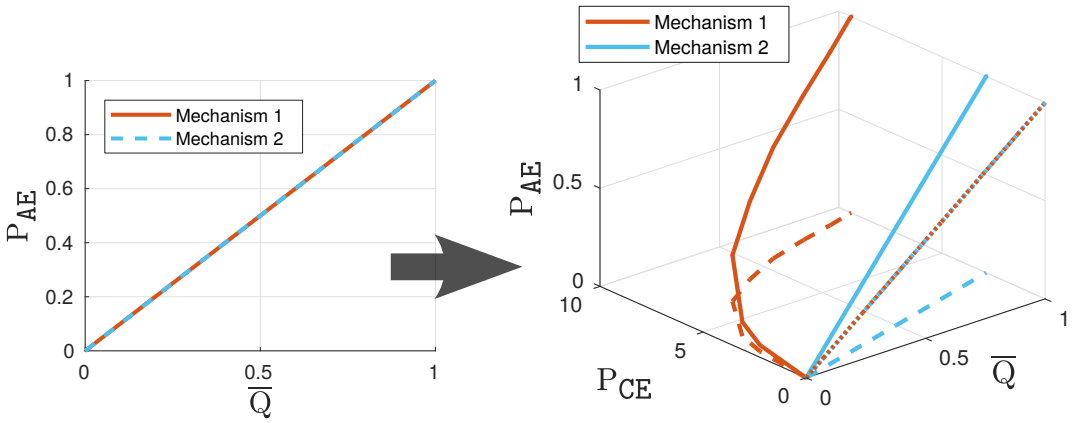


Figure 4.2: Two LPPMs that perform equally in the $\mathbf{P}_{\texttt{AE}}$ vs. $\overline{\text{Q}}$ plane, might behave very differently in practice. This is revealed by considering a multi-dimensional characterization of privacy.

measures the average error of the adversary's estimation in the most vulnerable output. When applied to the coin mechanism, this metric would reveal its privacy issue, since $\mathbf{P}_{\texttt{WC-AE}}(f_{\texttt{coin}}, \pi) = 0$.

On the other hand, the worst-case output conditional entropy, defined as

$$\mathbf{P}_{\texttt{WC-CE}}(f, \pi) = \min_{\substack{z \in \mathbb{R}^2 \\ f_Z(z) > 0}} \sum_{x \in \mathcal{X}} p(x|z) \cdot \log p(x|z)\,, \tag{4.25}$$

reveals the uncertainty the adversary has after observing z in the worst case (for the user). If there is any output value z that leaks a lot of information about the real location x (as it happens with every $z \neq z^*$ in the coin mechanism), this metric highlights it.

The metrics introduced throughout this section add additional dimensions to the privacy and quality loss evaluation procedure, revealing features not captured by the standard 2-dimensional approach based on the average error and the average loss. An example of this new characterization of privacy is shown in Fig. 4.2 where we show the performance of two LPPMs as a 3-D plot of $\mathbf{P}_{\texttt{AE}}$, $\mathbf{P}_{\texttt{CE}}$ and $\overline{\text{Q}}$, together with the projections in the $\mathbf{P}_{\texttt{AE}}$-$\overline{\text{Q}}$ and $\mathbf{P}_{\texttt{CE}}$-$\overline{\text{Q}}$ planes. In the next section, we show similar examples (albeit with 2-dimensional plots, for clarity) of particular location privacy preserving mechanisms.

4.5. Evaluation

In this section, we assess the performance of different location privacy-preserving mechanisms with respect to different privacy notions. Our experiments

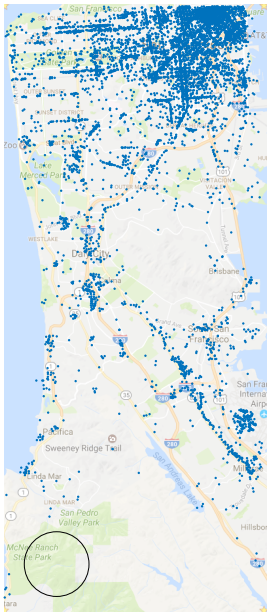


Figure 4.3: Points of interest in the San Francisco region taken from Gowalla dataset.

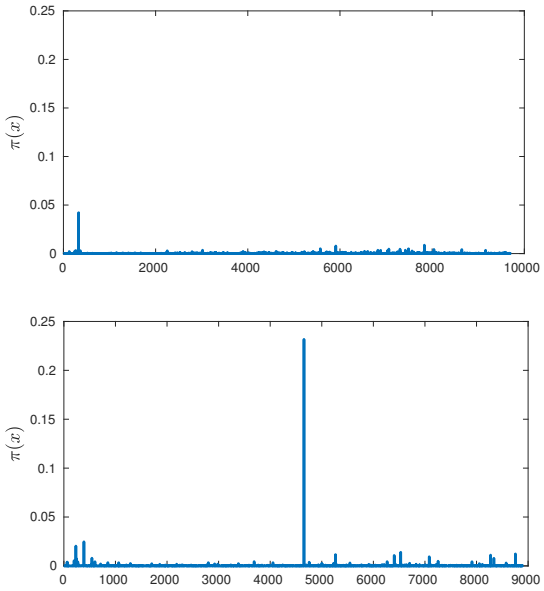


Figure 4.4: Priors π for Gowalla (left) and Brightkite (right) datasets.

confirm that relying on a single metric for evaluation can lead to an erroneous assessment of the privacy provided by an LPPM. We divide our evaluation into two parts. First, we consider the continuous scenario introduced in Section 4.2 and use real datasets to evaluate the performance of unbounded LPPMs, and of LPPMs that guarantee a maximum worst-case quality loss. Second, we consider a simpler scenario where the locations can only belong to a discrete set, and evaluate other defenses that have been proposed in the literature. All our experiments are performed using Matlab.[1]

4.5.1. Continuous Scenario

For this part of the evaluation, we consider that users are interested in querying about Points of Interest (PoIs) in a discrete set but they can report any point in $\mathbb{R}^2$ to the server (see Section 4.2). We also consider that the adversary performs her estimation in $\mathbb{R}^2$. We build the set of PoIs using the Gowalla[2] and Brightkite[3] real-world datasets. Following the approach of the finite domain evaluation in [67], we restrict the PoIs to a finite region of San Francisco area between the latitude coordinates (37.5395 and 37.7910) and longitude (-122.5153 and -122.3789). We choose the San Francisco area because it contains a big den-

[1] `https://www.mathworks.com/products/matlab.html`
[2] `https://snap.stanford.edu/data/loc-gowalla.html`
[3] `https://snap.stanford.edu/data/loc-brightkite.html`

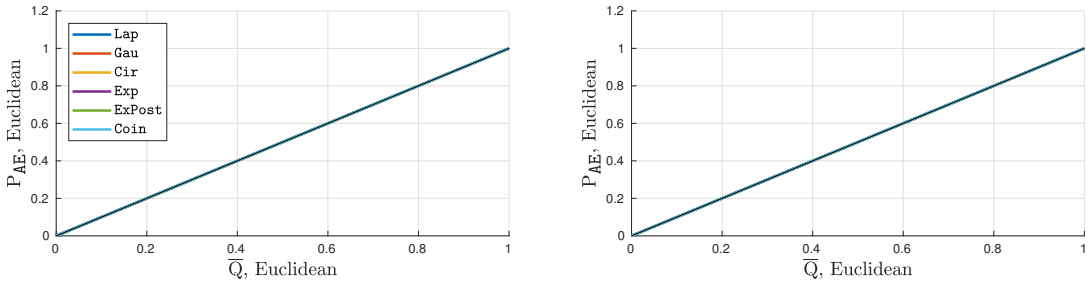


Figure 4.5: Average error vs. average quality loss for different unbounded LPPMs for Gowalla (left) and Brightkite (right) datasets.

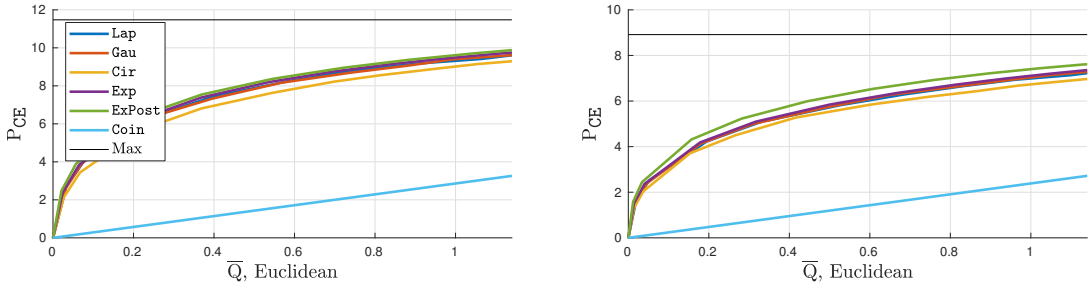


Figure 4.6: Conditional entropy vs. average quality loss for Gowalla (left) and Brightkite (right) datasets.

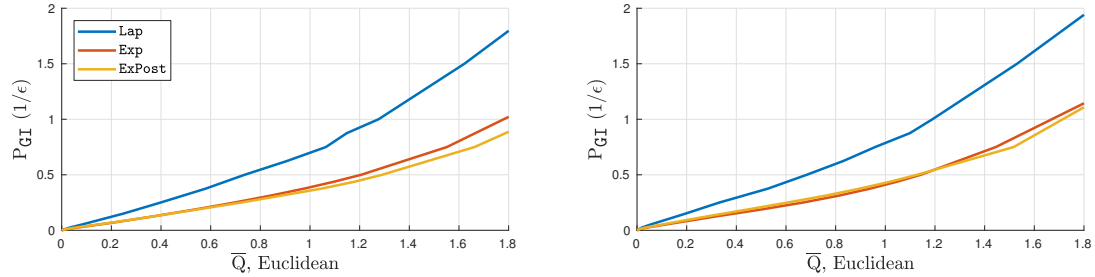


Figure 4.7: Geo-Ind Privacy $\mathbf{P}_{\mathtt{GI}}$ vs. average quality loss for Gowalla (left) and Brightkite (right) datasets.

sity of points of interest and a large number of user check-ins, which ensures that the data is rich and representative of what one would expect from users living in the area. On the other hand, considering a finite region allows us to evaluate LPPMs whose computational cost increases with the number of points of interest, such as the exponential and exponential posterior LPPMs. We transform the PoIs into Cartesian coordinates in kilometers using the Haversine formula with respect to the center of the region. We end up with $|\mathcal{X}| = 9\,701$ PoIs for Gowalla and $|\mathcal{X}| = 8\,898$ for Brightkite, distributed in an area of roughly $28\text{km} \times 12\text{km}$. As example, the distribution of PoIs for Gowalla is shown in Fig. 4.3. For each dataset, we compute the mobility profile π by counting how many users check-in on each point of interest and normalizing the resulting histogram. The obtained mobility profiles are shown in Fig. 4.4. We see that, in both datasets, there is a single

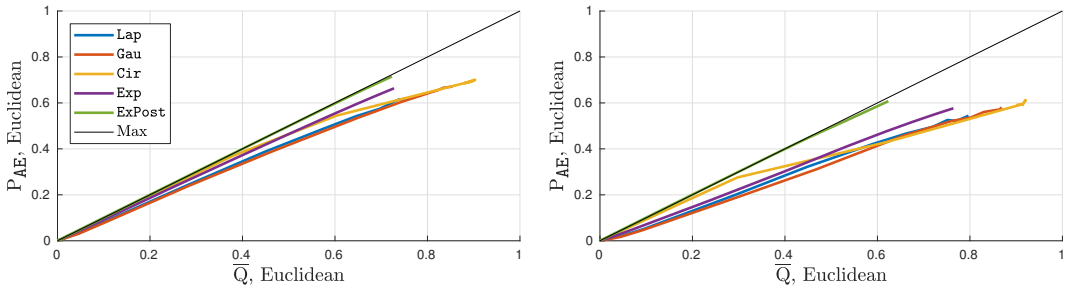


Figure 4.8: Average error vs. average quality loss for different bounded LPPMs.

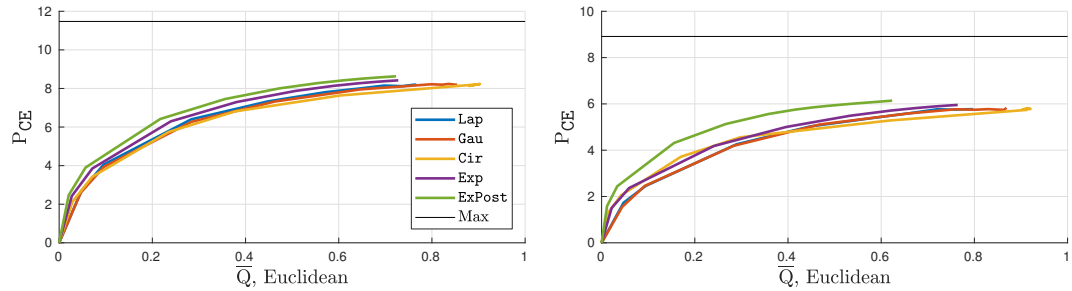


Figure 4.9: Conditional entropy vs. average quality loss for different bounded LPPMs.

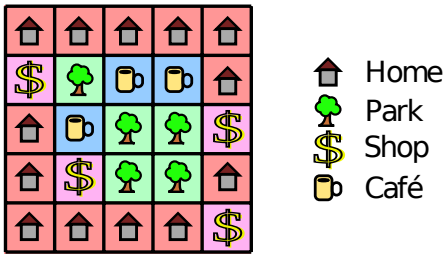


Figure 4.10: Semantic map of the discrete synthetic scenario.

point of interest x_{top} that draws a lot of attention from the users ($\pi(x_{\text{top}}) \approx 0.04$ in Gowalla and $\pi(x_{\text{top}}) \approx 0.23$ in Brightkite).

We evaluate six location-privacy preserving mechanisms, measuring their performance in terms of the average adversary error ($\mathbf{P}_{\texttt{AE}}$), conditional entropy ($\mathbf{P}_{\texttt{CE}}$) and geo-indistinguishability ($\mathbf{P}_{\texttt{GI}}$) for different values of average quality loss ($\overline{\text{Q}}$). We always use the Euclidean distance for the quality loss $d_Q(x, z) = ||x - z||_2$, and therefore the optimal remapping in (4.15) is obtained by computing the geometric median of the posterior. We compute this median using Weiszfeld's iterative method. We first evaluate the LPPMs without any bounds on their worst-case quality loss, and then imposing such constraint.

The first three LPPMs we evaluate consist in adding noise in the continuous plane and then remapping them. We generate this noise in polar coordinates,

sampling θ from a uniform distribution in $(0,\, 2\pi)$ and the radius r from a distribution specified below. Since for these algorithms we cannot find a closed form expression for $f(z|x)$, we evaluate them empirically. To this end we sample π to obtain x, we obtain z adding the noise and performing the remapping, and then we measure privacy according to each metric. We report averages over 5 000 repetitions. These LPPMs are:

- [`Lap`] **Planar Laplacian noise** plus remapping [67]. To generate the radius of the Laplace noise, we first sample p uniformly in the interval $(0, 1)$. Then, following [45], we set $r = \frac{1}{\epsilon}\left(W_{-1}\left(\frac{p-1}{e}\right) + 1\right)$ where W_{-1} is the -1 branch of the Lambert W function. We test different values of ϵ from 0.4km^{-1} to 40km^{-1}, so that the average loss varies between 0.05 and 5km.

- [`Gau`] **Bi-dimensional Gaussian noise** plus remapping. To generate Gaussian noise, we sample the radius from a Rayleigh distribution, varying its mean from 0.05 to 5km.

- [`Cir`] **Uniform circular noise** plus remapping. In this case, we sample the radius $r \in (0, R)$ from $f(r) = r/R^2$, where R is the maximum radius of the circle, which we vary from 0.075km to 7.5km. This ensures an average loss that varies between 0.05 and 5km.

Second, we evaluate three LPPMs that output values in a discrete set, whose conditional probability density functions $f(z|x)$ can be computed arithmetically. This allows us to exactly determine their privacy and quality loss performance. These LPPMs are:

- [`Coin`] **The coin mechanism,** explained in Sect. 4.3.2. We vary its average loss $\overline{\text{Q}}$ from 0 to 2.

- [`Exp`] **The Exponential mechanism** plus optimal remapping. The exponential mechanism is a general differential privacy technique that can be applied to provide geo-indistinguishability [69]. We set $\mathcal{Z} = \mathcal{X}$ and set a parameter b, then compute the probability of mapping each input x to an output z as $p(z|x) = a \cdot e^{-b \cdot d_Q(x,z)}$, where a ensures that $\sum_{z \in \mathcal{Z}} p(z|x) = 1$. Then, we apply an optimal remapping to the outputs of this function and obtain $f(z|x)$. In the experiments, we vary b from 0.4km^{-1} and 40km^{-1}.

- [`ExPost`] **Exponential posterior mechanism**, proposed in Section 4.4.1.2. In our experiments we set the discrete output alphabet of this algorithm to $\mathcal{Z} = \mathcal{X}$.

4.5.1.1. Results for Unbounded LPPMs (no Q^+ Constraint)

When the worst-case quality loss is not constrained, the optimal remapping ensures that all LPPMs are optimal in terms of average error, i.e., $\mathbf{P}_{\texttt{AE}} = \overline{Q}$ (Fig. 4.5). This shows that the optimal remapping applied to *any* LPPM achieves an optimal performance, whether it was Laplacian noise or a binary selection of a location such as `Coin`, as we proved in Sect. 4.3.

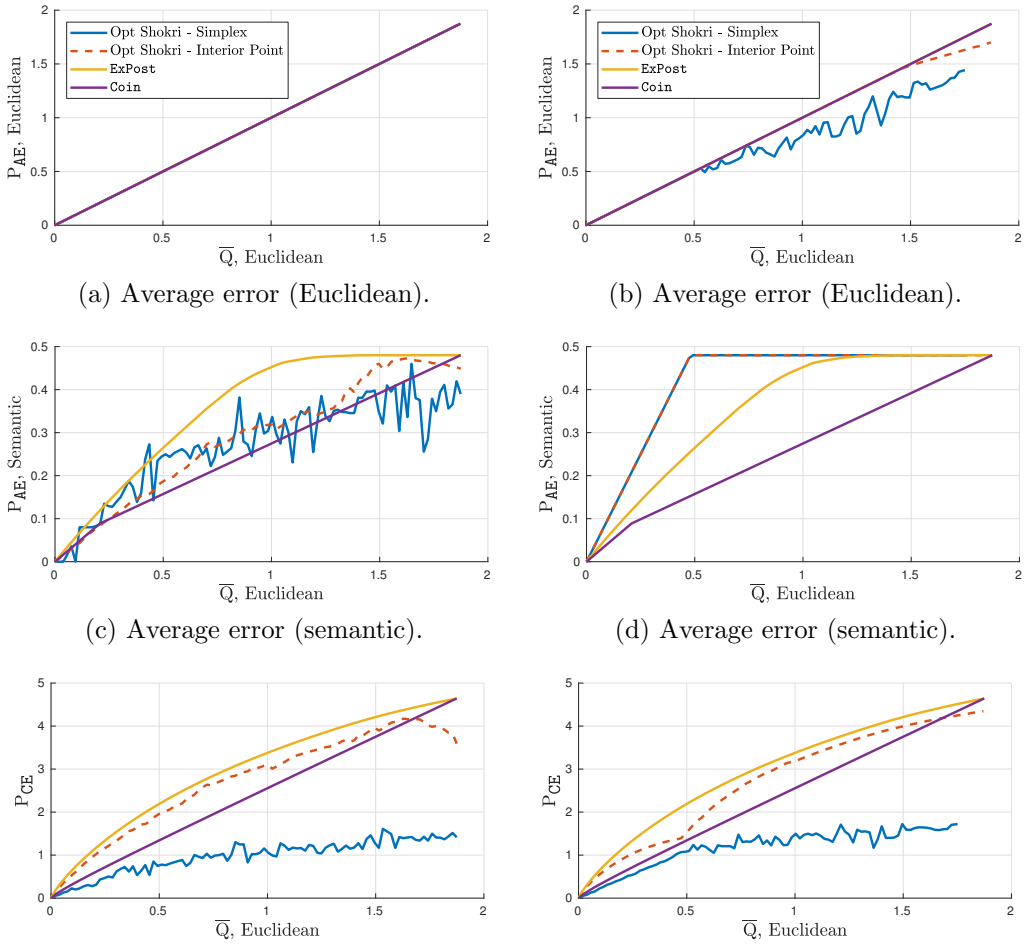
Figure 4.6 shows the LPPMs' performance in terms of conditional entropy $\mathbf{P}_{\texttt{CE}}$, where the horizontal black line represents the maximum entropy achievable, i.e., the entropy of the mobility profile π. Unsurprisingly, `ExPost` outperforms the rest of the LPPMs, as it is optimized with respect to this metric. The relative improvement of `ExPost` with respect to the other algorithms is slightly better in Brightkite than in Gowalla. This is due to the fact that in Brightkite the most frequent PoI is more popular than in Gowalla (see Fig. 4.4), and thus performing well in this location is crucial to achieve a good overall privacy level in Brightkite. The iterative structure of `ExPost` allows this LPPM to refine its performance and be more effective than the rest of the LPPMs around this PoI. We note, however, that this refinement comes at the price of an increase in computational cost. Overall, all the LPPMs achieve a similar performance in terms of conditional entropy, except for the coin, that performs poorly. This reinforces the critique in Sect. 4.3.2: even though `Coin` is optimal in terms of the average adversary error, measuring its performance in terms of conditional entropy reveals its privacy flaws.

Figure 4.7 shows the LPPMs' performance in terms of geo-indistinguishability $\mathbf{P}_{\texttt{GI}}(f)$ (we recall that $\mathbf{P}_{\texttt{GI}}(f) = 1/\epsilon$), only for `Lap`, `Exp` and `ExPost`, as these are the only algorithms that guarantee this property. As already seen in [67], the Laplace noise outperforms the exponential mechanism, and `ExPost` performs similar to the latter.

4.5.1.2. Results for Bounded LPPMs

We now impose a worst-case quality loss constraint of $Q^+_{\texttt{max}} = 1.5$km to the LPPMs (as a reference, we show a circle of radius 1.5km in Fig. 4.3). To implement this constraint in the LPPMs, we truncate their output at 1.5km and then apply the optimal remapping that respects the worst-case loss constraint. We do this by solving the problem in (4.15) with constraints. We do not evaluate the coin mechanism in this scenario, since it almost always violates the Q^+ constraint.

The results for the average adversary error as Euclidean distance are shown in Fig. 4.8. As expected, the LPPMs obtained after the remapping in this scenario are not necessarily optimal. We see that `ExPost` achieves a result that is close to the optimal LPPM in the unbounded case, while the other LPPMs achieve



(a) Average error (Euclidean).

(b) Average error (Euclidean).

(c) Average error (semantic).

(d) Average error (semantic).

(e) Conditional entropy.

(f) Conditional entropy.

Figure 4.11: Performance of Shokri et. al's algorithm optimized for the adversary error in terms of Euclidean distance (left) and semantic distance (right), compared to the coin mechanism and exponential posterior mechanism.

less average privacy. We conjecture this is due to the iterative nature of `ExPost`, that refines its performance, while the other LPPMs are not optimized regarding the worst-case loss constraint. Again, `ExPost` achieves a wider advantage in Brightkite for the same reason explained above.

Figure 4.9 shows the performance of the bounded LPPMs in terms of conditional entropy. The results are similar to those in the unbounded scenario, with `ExPost` outperforming the others with a slightly wider advantage in this case. As bounded LPPMs do not achieve geo-indistinguishability, we do not evaluate the performance with respect to this metric in this scenario.

4.5.2. Discrete Scenario

We now consider a simple synthetic scenario and evaluate the optimal LPPMs obtained following the method by Shokri et. al [46]. In this work, the authors propose a linear program that finds an LPPM f inside the polytope of optimal LPPMs for $\mathbf{P}_{\texttt{AE}}$ given a constraint $\overline{\mathrm{Q}}$, i.e., $f \in \mathcal{F}_{\mathrm{Q}}^{\texttt{opt}}$. This approach is very versatile, as it can be computed for any pair of distance functions $d_P(\cdot)$ and $d_Q(\cdot)$. We set our synthetic scenario under the assumptions of that work: the input and output alphabets are discrete and identical $\mathcal{X} = \mathcal{Z}$, and the adversary can only estimate locations inside that same alphabet $\hat{\mathcal{X}} = \mathcal{X}$. For simplicity, we consider that the set of locations in $\mathcal{X}$ are the centers of the cells that make a 5×5 square grid and assign a tag to each location that can be "Home", "Park", "Shop" or "Café", as depicted in Fig. 4.10. We consider that the mobility profile is uniform $\pi(x) = 1/25\,,\ \forall x \in \mathcal{X}$. We measure the point-wise loss as the Euclidean distance $d_Q(x, z) = ||x - z||_2$ and consider two point-wise metrics of privacy: the Euclidean distance and a semantic distance defined as the Hamming distance between tags, i.e., $d_P(x, z) = 0$ if $\mathrm{Tag}(x) = \mathrm{Tag}(z)$, and $d_P(x, z) = 1$ otherwise. This metric is similar to the semantic metric in [64]. The average error computed using this distance function represents the probability that an adversary guesses incorrectly the tag of x.

We evaluate `ExPost` and `Coin` together with the optimal LPPM proposed in [46]. For the latter, we solve the linear program to find optimal LPPMs in terms of maximizing $\mathbf{P}_{\texttt{AE}}$ using the Euclidean distance (Fig. 4.11, left column) and the semantic distance we defined (Fig. 4.11, right column). As expected, the optimal LPPMs (Shokri et. al) achieve the optimal privacy when evaluated using the adversary's error for which they are optimized (Figs. 4.11a and 4.11d), but not when evaluated against a different metric (Figs. 4.11c and 4.11b). `ExPost` and `Coin` achieve maximum privacy in terms of Euclidean distance, as before, but not in terms of semantic distance. This example emphasizes that optimizing an LPPM with respect to a privacy metric may provide very bad performance with respect to other privacy criteria.

This experiment also shows another important idea: even though the solutions of the linear program both achieve approximately the same performance in terms of average error (optimal in Figs. 4.11a and 4.11d, suboptimal in Figs. 4.11c and 4.11b), they exhibit a radically different behavior in terms of conditional entropy. Indeed, using the LPPM computed with the simplex algorithm (an LPPM at a vertex of $\mathcal{F}_{\mathrm{Q}}^{\texttt{opt}}$), the adversary has much less uncertainty about x on average than if the user had implemented an LPPM from the interior of the polytope. This difference in entropy is also what allows us to tell apart an LPPM such as `ExPost` from `Coin`. Note that the LPPM computed by solving the linear program with the simplex algorithm performs even worse than the coin in terms of entropy, illustrating the dangers of optimizing privacy in only one dimension.

4.6. Conclusions

In this chapter, we have demonstrated the problems of using a single privacy metric as indicator of the performance of location privacy preserving mechanisms. We have proven that there is more than one optimal LPPM in terms of maximizing the average adversary error for a given average quality loss, and that the family of LPPMs that fulfill such condition behave differently in terms of other privacy metrics. Thus, optimizing defenses with only one privacy metric in mind may lead to LPPMs that offer poor protection in other dimensions of privacy. To avoid selecting underperforming LPPMs we propose the use of complementary criteria to guide the choice. We provide two example auxiliary metrics: the conditional entropy and the worst-case loss. We propose an optimal LPPM with respect to the former, and provide means to implement LPPMs according to the latter.

We evaluate the LPPMs, comparing them to previous work, on two real datasets. Our experiments confirm two important ideas: first, that we cannot find an LPPM that performs optimally with respect to every privacy metric. Second, that even if an LPPM performs well in a particular metric it does not imply that it is necessarily beneficial for the user. Our findings reveal the need to take a step back in LPPM design to integrate privacy as a multi-dimensional notion, in order to avoid solutions that provide a false perception of privacy.

Appendix

4.A. Proof: Optimal LPPM by Optimal Remapping

We prove Theorem 4.3.2. In order to do this, first notice that, when $d_P(\cdot) \equiv d_Q(\cdot)$, the quality loss $\overline{\mathrm{Q}}$ is an upper bound of privacy $\mathbf{P}_{\mathtt{AE}}$:

$$\begin{aligned}\mathbf{P}_{\mathtt{AE}}(f,\pi) &= \int_{\mathbb{R}^2} \min_{\hat{x}\in\mathbb{R}^2} \left\{ \sum_{x\in\mathcal{X}} \pi(x)\cdot f(z|x)\cdot d_P(x,\hat{x}) \right\} dz \\ &\leq \int_{\mathbb{R}^2} \left\{ \sum_{x\in\mathcal{X}} \pi(x)\cdot f(z|x)\cdot d_Q(x,z) \right\} = \overline{\mathrm{Q}}(f,\pi)\,, \end{aligned} \tag{4.26}$$

Now, assume that $f' = f \circ g$, and therefore

$$z = \operatorname*{argmin}_{z'\in\mathbb{R}^2} \sum_{x\in\mathcal{X}} \pi(x)\cdot f'(z|x)\cdot d_Q(x,z')\,. \tag{4.27}$$

The optimal adversary estimation of x given z given in (4.4) can be written as

$$\hat{x}(z) = \operatorname*{argmin}_{\hat{x}\in\mathbb{R}^2} \sum_{x\in\mathcal{X}} \pi(x)\cdot f'(z|x)\cdot d_P(x,\hat{x})\,. \tag{4.28}$$

We see that since $d_P(\cdot) \equiv d_Q(\cdot)$ the optimal adversary estimation is doing nothing, i.e., $\hat{x}(z) = z$. This implies that $\mathbf{P}_{\mathtt{AE}}(f',\pi) = \overline{\mathrm{Q}}(f',\pi)$, and since we have achieved the upper bound on privacy given in (4.26), f' is optimal.

4.B. Proof: Geo-indistinguishability of the Exp. Posterior Mechanism.

We recall that the geo-indistinguishability guarantee requires the following condition to be fulfilled (now written for discrete LPPMs, where $p(z|x)$ denotes

the probability of reporting z when the original location is x):

$$p(z|x) \leq e^{\epsilon \cdot d_P(x,x')} \cdot p(z|x')\,, \quad \forall x, x' \in \mathcal{X},\, z \in \mathcal{Z}\,, \tag{4.29}$$

where $d_P(x, x')$ is the Euclidean distance.

The last iteration of the `ExPost` algorithm in 4.4.1.2 returns an LPPM that can be written for a particular input x and output z as

$$p(z|x) = \begin{cases} \frac{P_Z(z) \cdot e^{-b \cdot d_Q(x,z)}}{\sum_{z' \in \mathcal{Z}} P_Z(z') \cdot e^{-b \cdot d_Q(x,z')}} & \text{if } P_Z(z) > 0\,, \\ 0\,, & \text{if } P_Z(z) = 0\,. \end{cases} \tag{4.30}$$

where $d_Q(x, z)$ is the Euclidean distance. In the second case, the geo-indistinguishability guarantee is trivially achieved since given any pair of input locations $x, x' \in \mathcal{X}$, $p(z|x) = p(z|x') = 0$. For the first case, we use the triangular inequality $d_Q(x, z) + d_Q(x', z) \geq d_Q(x, x')$ to write

$$\begin{aligned} p(z|x) =& \frac{P_Z(z) \cdot e^{-b \cdot d_Q(x,z)}}{\sum_{z' \in \mathcal{Z}} P_Z(z') \cdot e^{-b \cdot d_Q(x,z')}} && (4.31) \\ \leq& \frac{P_Z(z) \cdot e^{b \cdot d_Q(x,x')} \cdot e^{-b \cdot d_Q(x',z)}}{\sum_{z' \in \mathcal{Z}} P_Z(z') \cdot e^{-b \cdot d_Q(x,z')}} && (4.32) \\ \leq& \frac{P_Z(z) \cdot e^{b \cdot d_Q(x,x')} \cdot e^{-b \cdot d_Q(x',z)}}{\sum_{z' \in \mathcal{Z}} P_Z(z') \cdot e^{-b \cdot d_Q(x,x')} \cdot e^{-b \cdot d_Q(x',z')}} && (4.33) \\ =& \frac{P_Z(z) \cdot e^{-b \cdot d_Q(x',z)}}{\sum_{z' \in \mathcal{Z}} P_Z(z') \cdot e^{-b \cdot d_Q(x',z')}} \cdot e^{2b \cdot d_Q(x,x')} && (4.34) \\ =& e^{2b \cdot d_Q(x,x')} \cdot p(z|x')\,, && (4.35) \end{aligned}$$

which satisfies the geo-indistinguishability for $\epsilon = 2b$ or $\mathbf{P}_{\mathtt{GI}} = 1/2b$, if $d_Q(\cdot)$ is the Euclidean distance. This concludes the proof.

Chapter 5

Rethinking Location Privacy for Unknown Mobility Behaviors

5.1. Introduction

In the previous chapter, we designed and evaluated Location Privacy Preserving Mechanisms (LPPMs) using different privacy metrics. We assumed that user mobility can be characterized by the *mobility profile*, and considered that this profile is known a-priori (in our case, we extracted it from the same data that we used for evaluation). Training the users' mobility model using evaluation data is not new, but inherited from [51], and it is a typical procedure in most of the location privacy literature [46, 64, 64, 65, 65, 66, 66, 70, 90–92].

In practice, the LPPM designer trains the user mobility model on *past* data (since *future* data is not available). However, gathering mobility data that is sufficient, up-to-date, and truly representative of a particular user's behavior is complicated. In most cases, user behavior is to some degree unknown and thus LPPMs hardwired on (past) training data will not be optimal in practice. Also, evaluating LPPMs on the same data used for their design is highly unrealistic, and does not give a real sense of the privacy that these mechanisms provide.

Chatzikokolakis et al. have recently acknowledged part of this problem in [67], where they claim that a fair assessment of LPPMs requires the separation between the *training* dataset used for design, and the *testing* dataset used for evaluation. Yet, their design strategy, as the rest of the previous works, hardwires the training mobility model into the mechanism and they do not quantify how much privacy

This chapter is adapted with permission from IEEE: Simon Oya, Carmela Troncoso, and Fernando Pérez-González. Rethinking location privacy for unknown mobility behaviors. In IEEE European Symposium on Security and Privacy (EuroS&P)*, IEEE 2019.*

is lost in practice when the users' mobility characteristics differ from the training data.

In this chapter, we aim at understanding the privacy loss associated to this discrepancy between design and deployment phases. We study both sporadic cases, where users query the Location Based Service (LBS) occasionally and thus their location is independent from previous LBS uses; and continuous cases, where users' actual location at a certain time depends on previously visited locations. We find that, since the design strategies in previous works *hardwire* the training information on the LPPMs they produce, they cannot adapt to behavioral patterns not available in the training data. We empirically show that, indeed, previous analyses overestimate the protection of the optimal LPPMs when they are evaluated on mobility profiles different from the training data.

In response to this problem, we introduce a new design strategy that builds on what we call *blank-slate models* for user mobility. Contrary to hardwired models, blank-slate models do not fix their parameters based on training data, but learn these parameters as they observe the user behavior. We take the particular case of sporadic location privacy and leverage a blank-slate model to build a new family of defenses that we call Profile Estimation-Based LPPMs (PEB-LPPMs). Like traditional LPPMs, these mechanisms are initialized with training data. However, as the user queries the LBS, they adapt their parameters. Thus, they are more adequate for those users whose behavior is not well-represented in the training data. We empirically compare PEB-LPPMs with state-of-the-art LPPMs using real data. Our evaluation confirms that PEB-LPPMs are more effective than traditional hardwired models when the testing data cannot be fully characterized a-priori by the training data.

To summarize, our contributions are:

- We empirically show that hardwiring the characteristics of a dataset into Location Privacy Preserving Mechanisms [46, 51, 64–67, 90–93] yields mechanisms that do not adequately protect users whose behavior deviates from that observed in training.
- We propose *blank-slate models* for user mobility in location privacy. Contrary to hardwired models, these models treat the user mobility as an *unknown variable* that is learned a-posteriori as the user queries the LBS. Therefore, they enable the design of LPPMs that are effective when the user behavior changes with respect to the one observed when designing the mechanism.
- We leverage a blank-slate sporadic mobility model to develop a new LPPM design technique, that we call Profile Estimation-Based (PEB). PEB-LPPMs adapt to the user behavior by performing a Maximum Likelihood

Estimation (MLE) of the mobility profile given past observations, and are suitable for both sporadic and non-sporadic location protection.

- We compare PEB-LPPMs with optimal state-of-the-art designs developed using hardwired sporadic and Markov models. PEB-LPPMs always outperform optimal sporadic hardwired LPPMs, and sometimes they even outperform optimal LPPMs based on Markov models if the training data does not correctly capture the mobility behavior of the users of the testing set.

- To carry out this comparison we extend efficient remapping techniques used in optimal sporadic LPPMs [67] to build optimal non-sporadic Markov-based LPPMs [91, 93]. This considerably reduces the computational cost of building non-sporadic LPPMs and allows us to evaluate them empirically.

The rest of the chapter is organized as follows. In the next section, we introduce our system model and notation, as well as the evaluation framework that we use in the chapter. Section 5.3 presents the sporadic and Markov mobility models. Then, in Sect. 5.4, we explain how previous works use these mobility models, hardwired on training data, to build optimal LPPMs. We train and evaluate these optimal LPPMs with real data in Sect. 5.5, showing that there is a gap between their theoretical performance and their actual performance in the testing set. We introduce blank-slate models and our technique to develop PEB-LPPMs in Sect. 5.6, and evaluate it in Section 5.7. Finally, Sect. 5.8 summarizes related work and Sect. 5.9 concludes.

5.2. Overview of the Location Privacy Problem

In this section, we first provide an abstraction of the location privacy problem and introduce our notation. Then, we present our framework for design and evaluation of LPPMs.

5.2.1. Problem Statement and Notation

As in the previous chapter, we consider the scenario where an individual, the *user*, sends queries to an LBS provider and receives responses with the information she desires. We consider that there is a passive *adversary* observing the locations inside the user queries. This adversary can be an honest-but-curious LBS or an eavesdropper. The adversary's her goal is to infer private information from the locations in user queries [94, 95]. To protect herself the user obfuscates her locations using an LPPM, and sends these fake locations in the queries. By doing so, the user trades in quality of service for privacy.

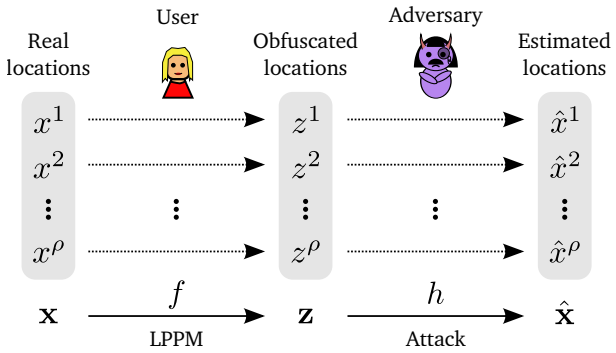


Figure 5.1: Abstraction of the location privacy problem.

We illustrate the location privacy problem in Fig. 5.1. The model is similar to the one presented in the previous chapter, but takes into account more than a single query of the user. We use ρ to denote the total number of queries sent by the user to the LBS, and refer to each query by its query number $r \in \{1, 2, \cdots, \rho\}$. We use $x^r \in \mathcal{X}$ to denote the real location associated with the r-th query, i.e., the location the user wants to query about. We use $\mathbf{x} \doteq [x^1, \cdots, x^\rho] \in \mathcal{X}^\rho$ to denote the vector of all the real locations, and $\mathbf{x}^r \doteq [x^1, \cdots, x^r] \in \mathcal{X}^r$ to denote the vector of all the real locations up to query number r. Likewise, we use $z^r \in \mathcal{Z}$ to denote the r-th fake location reported and define the vectors $\mathbf{z}$ and $\mathbf{z}^r$. The real and fake locations are also called input and output locations respectively. Finally, we use $\hat{x}^r \in \hat{\mathcal{X}}$ to denote the adversary's estimation of x^r.

In this chapter, we assume that $\mathcal{X}$, $\mathcal{Z}$ and $\hat{\mathcal{X}}$ are discrete sets of locations (i.e., the users can only report locations in a grid). We do this for computational simplicity and for compatibility with previous proposals [46, 91, 93]. However, all of our findings can be extended to other scenarios (e.g., $\mathcal{Z} = \mathbb{R}^2$ is the plane [92], $\mathcal{Z}$ is a discrete set of cloaking regions [56], or a powerset of points of interest [90]). We use $x_1, x_2, \cdots, x_{|\mathcal{X}|}$ to denote each of the discrete locations in $\mathcal{X}$. Finally, we use p generally to denote the probability mass function of a discrete random variable, or the probability density function when the variable is continuous. $\mathrm{E}\{\cdot\}$ denotes the expectation.

Now, we explain how real, obfuscated, and estimated locations are generated. The real locations $\mathbf{x}$ are chosen by the user as she queries the LBS. In some scenarios, the user makes a *sporadic* usage of the LBS (e.g., location check-in, location-tagging, or applications for finding nearby points-of-interest or friends). This means that the real locations of two queries (e.g., x^r and x^s, with $r \neq s$) are not temporally dependent. In other scenarios, however, the location of the user in consecutive check-ins is correlated (e.g., a user that reports her location frequently, such as running apps or WhatsApp's live location sharing).

In order to generate obfuscated locations $\mathbf{z}$ from the real locations $\mathbf{x}$ the user

employs an LPPM f. We study the *online* location privacy setting, in which the user expects to get the service from the LBS right away. In this case, the LPPM is modeled as a probabilistic function that maps a real location $x^r \in \mathcal{X}$, and possibly other information available to the user up to that point (i.e., $\mathbf{x}^{r-1}$ and $\mathbf{z}^{r-1}$), to a value $z^r \in \mathcal{Z}$. We use f to denote the probability density function that characterizes the LPPM. Hence, we can write $p(\mathbf{z}|\mathbf{x})$ as

$$p(\mathbf{z}|\mathbf{x}) = \prod_{r=1}^{\rho} p(z^r|\mathbf{z}^{r-1}, \mathbf{x}) = \prod_{r=1}^{\rho} f(z^r|\mathbf{z}^{r-1}, \mathbf{x}^r)\,, \tag{5.1}$$

where the first equality is the chain rule of probability and the second equality reflects the online setting assumption, i.e., the user generates z^r given $\mathbf{x}^r$ and $\mathbf{z}^{r-1}$, but independently of future locations x^{r+1}, x^{r+2}, etc. We also refer to f as the *obfuscation mechanism*.

Finally, the adversary generates the estimated locations using an attack h. We assume that the adversary knows the obfuscation mechanism f and she uses it to design her attack h. We treat h as a deterministic function that takes a vector of obfuscated locations $\mathbf{z}^r$ and produces an estimate $\hat{x}^s$ of a (possibly past) real location x^s ($s \leq r$). We use $\hat{x}^s(\mathbf{z}^r)$ to denote the estimate produced from $\mathbf{z}^r$ using h. We do not consider randomized attacks, since the goal of the adversary is to choose her estimation so as to minimize a specific privacy metric, which can be achieved with deterministic attacks.

LPPM types: Depending on how much information they use to generate obfuscated locations, LPPMs can offer stronger privacy guarantees at the cost of introducing complexity in the design. In this chapter we study the following LPPM types that can accommodate all previous proposals in the literature:

1. *Full LPPMs* are the most generic LPPM in the online location privacy setting (see (5.1)), i.e., $f(z^r|\mathbf{z}^{r-1}, \mathbf{x}^r)$. They generate each obfuscation location z^r (perhaps randomly) using all the information available to the user, i.e., the previous and current input locations $\mathbf{x}^r$, and the previously released obfuscated locations $\mathbf{z}^{r-1}$.

2. *Output-based LPPMs*, $f(z^r|\mathbf{z}^{r-1}, x^r)$, generate the obfuscated location using only the current real location x^r and all the previous obfuscated locations $\mathbf{z}^{r-1}$. These are a sub-type of full LPPMs.

3. *Memoryless LPPMs*, $f(z^r|x^r)$, generate each obfuscated location using the current real location x^r only. These are a sub-type of output-based LPPMs.

We note that the framework in [51] considers LPPMs of the full type in its theoretical setup, but the evaluation studies only memoryless LPPMs. Memoryless LPPMs are used in sporadic location privacy and works that consider a single

Table 5.1: Summary of notation

Symbol	Meaning
ρ	Total number of queries.
x^r	Real location of the user in the r-th query.
z^r	Obfuscated location of the user in the r-th query.
$\mathbf{x}^r$ (or $\mathbf{z}^r$)	Vector of real (or obfuscated) locations up to query r.
$\mathbf{x}$ (or $\mathbf{z}$)	Vector of all real (or obfuscated) locations.
$\mathcal{X}$ (or $\mathcal{Z}$)	Set of all possible real (or obfuscated) locations.
h	Adversary's attack.
$\hat{x}^s(\mathbf{z}^r)$	Adversary's estimate of the real location x^s using $\mathbf{z}^r$.
f	LPPM or obfuscation mechanism (pdf that generates z^r).
$f(z^r\|\mathbf{z}^{r-1}, \mathbf{x}^r)$	Full LPPM.
$f(z^r\|\mathbf{z}^{r-1}, x^r)$	Output-based LPPM.
$f(z^r\|x^r)$	Memoryless LPPM.
$d_Q(x^r, z^r)$	Quality loss when reporting z^r given x^r.
$\overline{\mathrm{Q}}(f, s)$	Average quality loss metric at query number r (5.2).
$d_P(x^r, \hat{x}^r)$	Adv. error when the adversary estimates x^r as $\hat{x}^r$.
$\mathbf{P}_{\mathtt{AE}}(f, h, r, s)$	Avg. adv. error of $\hat{x}^s$ given $\mathbf{z}^r$ and attack h (5.5).

location release [46, 66, 67, 72, 90, 92]. Output-based LPPMs are typically used in non-sporadic location privacy works [91, 93] and, to the best of our knowledge, no optimal full-LPPM has been proposed due to the computational complexity inherent to its design.

The notation used in the chapter is summarized in Table 5.1.

5.2.2. Design and Evaluation Framework

We now describe a framework that instantiates the abstraction above. This framework extends ideas from [51, 67]. It consists of two steps: the design step, where the user designs the LPPM f; and the evaluation step, where the performance of f is evaluated empirically. The framework is represented in Fig. 5.2.

Design Step: In this step, the user studies the location privacy problem and builds the LPPM f. We assume that the user has access to a *training set*. She derives her design according to some performance requirements, in terms of privacy and utility metrics (e.g., maximizing privacy while keeping the utility level above some bounds). Also, the user does not know the adversary's attack h, so she designs the LPPM considering a worst case adversary. In order to compute the privacy and utility metrics, the user needs a model for the joint distribution

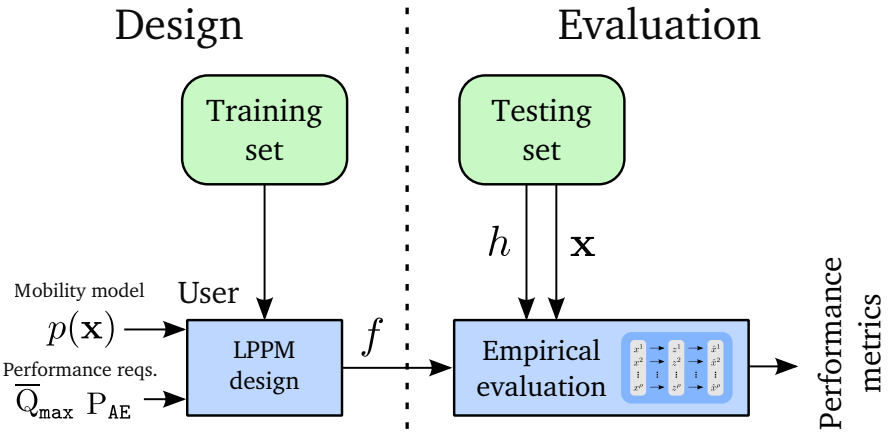


Figure 5.2: LPPM design and evaluation framework.

$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}) \cdot p(\mathbf{z}|\mathbf{x})$. The first term, $p(\mathbf{x})$, is the joint distribution of the real locations of the user. The user derives this distribution by training her mobility model with the training set information. The second term, $p(\mathbf{z}|\mathbf{x})$, is determined by the LPPM f, as in (5.1).

Evaluation Step: In this step, the performance of f against one or more attacks h is assessed empirically using a *testing set*. Following Kerckhoffs principle, we assume that the adversary knows the user LPPM, and uses an optimal attack, i.e., an attack that minimizes the privacy metric. To develop the worst-case attack we assume that the adversary knows mobility statistics about the testing set (e.g., the actual probability distribution of $\mathbf{x}$), a common assumption in related works [46, 51, 91, 92].

The testing set contains real traces of locations $\mathbf{x}$ from a location privacy dataset. The outputs $\mathbf{z}$ are probabilistically generated using f and $\mathbf{x}$. Then, the estimations $\hat{x}^s$ $(s = 1, 2, \cdots)$ are calculated using h and $\mathbf{z}$. The privacy and utility performance of the LPPM is assessed empirically based on $\mathbf{x}$, $\mathbf{z}$ and $\hat{x}^s$.

Note that there is a fundamental difference between the design and the evaluation steps, regarding the treatment of the real locations $\mathbf{x}$. The design step is carried out by studying the problem *analytically*, and this is done by assuming a particular mobility model for the real locations $p(\mathbf{x})$. The evaluation step, on the contrary, is carried out *empirically* with real samples of $\mathbf{x}$. Ideally, the user wants her mobility model to closely resemble her real behavior, so that her theoretical analyses translate well into practice. However, finding a realistic model for $\mathbf{x}$ is a very complicated task due to the unpredictability and complexity of user behavior. Notice that this is not an issue for the generation of $\mathbf{z}$. This is because these samples are generated by $p(\mathbf{z}|\mathbf{x})$, which is completely characterized by the obfuscation mechanism and is the same in the design and evaluation steps.

Main Differences with Previous Work: This framework takes ideas from the

literature, but also adds some contributions. The framework by Shokri et al. [51] considers an adversary that designs her attack based on the evaluation data, but does not evaluate LPPMs designed to maximize privacy. The separation between training and testing data is considered for the first time in [67], but there is no quantification of the privacy loss associated to users' whose mobility profiles diverge from the training data.

In this chapter, we integrate the training/testing separation as part of the framework. We also consider the selection of a model for the real locations $p(\mathbf{x})$ as a crucial part of the designing step, which was considered as given by previous works [46, 51, 64–67, 90–93]. Finding a suitable theoretical model for the user mobility $p(\mathbf{x})$ and fitting it to the training data is part of the LPPM design process. However, we cannot take for granted that the actual locations of the user in practice $\mathbf{x}$ will follow the theoretical model that she considered for design, and thus the performance of the LPPM in practice might differ from the theoretical performance.

5.2.3. Performance Metrics

We quantify the performance of LPPMs using privacy and utility (or quality loss) metrics. Even though in Chapter 4 we showed that a fair assessment of LPPM performance should be carried out by considering many privacy metrics, for the purposes of this chapter it is enough to use only the average quality loss as utility metric, and the average adversary error as privacy metric. These metrics are the most popular in the user-centric location privacy literature [45, 46, 51, 64–67, 91–93]. We now define these metrics, and explain how to compute them analytically given a model of $p(\mathbf{x})$, and empirically given samples of pairs $(\mathbf{x}, \mathbf{z})$. Later, in Section 5.7.3 we explain why the improvements that we achieve in terms of average adversary error would also apply to other privacy metrics.

Utility Metric: Average Quality Loss. The average quality loss measures how much quality the user loses on average by reporting obfuscated locations instead of real ones [45,46,65–67,91–93]. Let $d_Q(x, z)$ be a point-to-point *distance function* that measures the loss incurred by revealing z when the real location is x. The average loss at query r given LPPM f is

$$\overline{\mathrm{Q}}(f, r) \doteq \mathrm{E}\left\{d_Q(x^r, z^r)\right\} , \tag{5.2}$$

where the expectation is taken over realizations of x^r and z^r. Given a distribution $p(\mathbf{x})$, we can compute this metric theoretically as

$$\overline{\mathrm{Q}}^{\texttt{theo}}(f, r) = \sum_{x^r \in \mathcal{X}} \sum_{z^r \in \mathcal{Z}} p(x^r) \cdot p(z^r | x^r) \cdot d_Q(x^r, z^r) , \tag{5.3}$$

where $p(x^r)$ and $p(z^r|x^r)$ can be obtained analytically from $p(\mathbf{x})$ and $p(\mathbf{z}|\mathbf{x})$.

Empirically, we can compute this metric by averaging the distance between x^r and z^r over multiple simulations, i.e.,

$$\overline{\mathrm{Q}}^{\mathtt{prac}}(f, r) = \mathrm{E}_{\mathtt{emp}}\{d_Q(x^r, z^r)\}\,, \tag{5.4}$$

where $\mathrm{E}_{\mathtt{emp}}\{\cdot\}$ denotes the empirical mean.

The typical choice for the distance function $d_Q(\cdot)$ is the Euclidean distance. However, $d_Q(\cdot)$ can be tailored to the particular application where we want to provide location privacy. For example, in an application to find nearby points of interest within a city, the Manhattan distance is appropriate to measure the walking distance to go from x to z. In that case, $\overline{\mathrm{Q}}$ would represent the average amount of extra meters that the user has to walk to reach the desired point of interest. In a ride-sharing app, however, $d_Q(x, z)$ can represent the extra time or money that the user loses by reporting z instead of her real location x. We can also use semantic metrics based on the location tags of x and z, etc.

Privacy Metric: Average Adversary Error. The average adversary error is defined as the mean error incurred by an adversary that estimates the user real locations using an attack h [45, 46, 51, 64–67, 91–93]. Let $d_P(x, \hat{x})$ be a function that quantifies how much privacy the user has when her real location is x and the location estimated by the adversary is $\hat{x}$. Typically, $d_P(\cdot)$ is the Euclidean distance, but it can adapted to a particular application. Consider that the adversary has observed r outputs ($\mathbf{z}^r$) and wants to estimate the location x^s with $s \leq r$. For this, she uses an attack h that produces an estimation $\hat{x}^s(\mathbf{z}^r)$. The average adversary error at query r regarding x^s can be defined as

$$\mathbf{P}_{\mathtt{AE}}(f, h, r, s) \doteq \mathrm{E}\left\{d_P(x^s, \hat{x}^s(\mathbf{z}^r))\right\}\,, \tag{5.5}$$

where the expectation is taken over x^s and $\mathbf{z}^r$ (the attack is deterministic, i.e., $\hat{x}^s$ is a function of $\mathbf{z}^r$). Given a mobility model $p(\mathbf{x})$, this metric can be computed analytically as

$$\mathbf{P}_{\mathtt{AE}}^{\mathtt{theo}}(f, h, r, s) = \sum_{\mathbf{x}^r \in \mathcal{X}^r} \sum_{\mathbf{z}^r \in \mathcal{Z}^r} p(\mathbf{x}^r) p(\mathbf{z}^r | \mathbf{x}^r) d_P(x^s, \hat{x}^s(\mathbf{z}^r))\,. \tag{5.6}$$

Empirically, for each realization of $\mathbf{x}$ and $\mathbf{z}$, we obtain the adversary estimation $\hat{x}^s(\mathbf{z}^r)$, and then compute the average adversary error as

$$\mathbf{P}_{\mathtt{AE}}^{\mathtt{prac}}(f, h, r, s) = \mathrm{E}_{\mathtt{emp}}\{d_P(x^s, \hat{x}^s(\mathbf{z}^r))\}\,. \tag{5.7}$$

We acknowledge that there are other privacy metrics, e.g., the conditional entropy [92] and geo-indistinguishability [45]. In our empirical evaluation in Sect. 5.7.3 we discuss how our findings affect those metrics.

5.3. Mobility Models for LPPM Design

As we explained before, in order to design LPPMs, the user needs to assume a model that characterizes her mobility behavior, i.e., a model for $p(\mathbf{x})$. In this section, we explain the main mobility models assumed in the literature: the *sporadic* mobility model, and the *Markov* model (non-sporadic). We do not claim that there is a *correct* mobility model for $p(\mathbf{x})$ that the user should follow. However, it is true that LPPMs optimized for a certain model will perform better when the actual user location traces follow such model. In other words, models that are closer to real behavior are more *useful*.

5.3.1. Sporadic Model

The sporadic location privacy model assumes that the real locations of the user in two different queries, i.e., x^r and x^s, are not temporally dependent. As we argued before, this makes sense in some scenarios where the user requests information from the LBS infrequently (e.g., a user that queries for the weather in her area is not likely to perform the another query in a short period of time).

The sporadic model characterizes $p(\mathbf{x})$ using a parameter called the , denoted by π (Fig. 5.3, left, and Fig. 5.4). The mobility profile is an abstraction that represents the long-term user behavior, i.e., the probability with which the user visits each location $x \in \mathcal{X}$. Thus, given π, we can write

$$p(\mathbf{x}|\pi) = \prod_{r=1}^{\rho} p(x^r|\pi) = \prod_{r=1}^{\rho} \pi(x^r)\,, \tag{5.8}$$

where we have used $\pi(x)$ to denote the probability that the user's real location is x given the profile π.

This model has been widely used in the literature [46, 65, 67, 92], mainly for its simplicity: using the fact that two check-ins x^r and x^s are independent allows the user to design LPPMs that only need the current input x^r to generate the next output z^r.

5.3.2. Continuous Model: Markov

In some scenarios, the sporadic model for user mobility is not appropriate. For example, when a user queries the LBS continuously (e.g., live location sharing in social networks), we cannot assume that the location x^{r+1} is independent of the previous one x^r (e.g., because physical constraints such as the user speed or

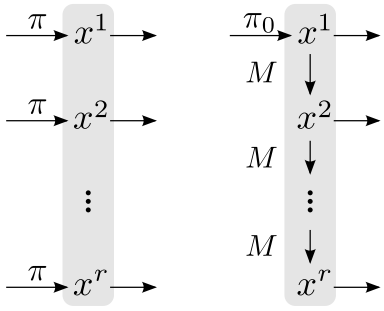


Figure 5.3: Sporadic (left) and Markov (right) models of user mobility.

	Meaning
π	Mobility profile $p(x^r\|\pi) = \pi(x^r)$
π_0	Initial mob. prof. $p(x^1\|\pi_0) = \pi_0(x^1)$
M	Transition matrix. $p(x^{r+1}\|x^r, M) = M(x^{r+1}\|x^r)$

Figure 5.4: Notation and meaning of the model parameters.

roads existence and direction). In those cases, continuous models that specify the dependencies between the real locations are more adequate to design LPPMs.

The most typical model in this scenario is the Markov model. As its name suggests, this model characterizes x^r as a Markov chain. More specifically, Markov models are defined by two parameters: an initial mobility profile π_0, and a *transition matrix* M (Fig. 5.3, right, and Fig. 5.4). The initial profile models the probability of the first location of the user, i.e., $p(x^1|\pi_0) = \pi_0(x^1)$. The transition matrix M is a $|\mathcal{X}| \times |\mathcal{X}|$ matrix whose (i,k)-th element characterizes $p(x_i^r|x_j^{r-1})$, regardless of $r > 1$. We use $M(x^r|x^{r-1})$ to denote the probability that the user transitions from location x^{r-1} to x^r according to the matrix M. The probability of a trace $p(\mathbf{x})$ according to the Markov model is thus

$$p(\mathbf{x}|\pi_0, M) = \prod_{r=1}^{\rho} p(x^r|\mathbf{x}^{r-1}, \pi_0, M) = \pi_0(x^1) \cdot \prod_{r=2}^{\rho} M(x^r|x^{r-1})\,. \tag{5.9}$$

The Markov model has been widely used in non-sporadic location privacy, due to its simplicity [91,93]. Note that in the Markov model, the user's mobility behavior only depends on her current location, and not the past trace. It is possible to define more complicated models for continuous location release (e.g., characterize $p(x^r|\mathbf{x}^{r-1})$), but since these are rarely used, we do not consider them in this chapter.

5.3.3. Hardwiring Training Data into the Mobility Model

After the user chooses one model to design her LPPM, she has to decide how to estimate the parameters of that model (i.e., π in the sporadic model, π_0 and M in the Markov model). In the literature, to the best of our knowledge, all of the proposals rely on some training information to determine these parameters [46,51, 64,65,67,91–93,96]. After this training phase, the model parameters remain fixed during the evaluation. We call the models that are built in this way *hardwired*

models. Hardwired models are tailored to the training data data *a-priori* during training, and their parameters are never updated or adapted for users that deviate from the training data behavior. Therefore, the LPPMs designed with these models will be optimal in practice if the users' behavior is perfectly captured by the training data. If this is not the case, *or if the training data is insufficient or nonexistent*, it is reasonable that the LPPMs designed with hardwired models will perform worse than expected. We confirm this conjecture later in Section 5.5.

5.4. LPPM Design in Hardwired Models

In this section, we overview previous approaches to design LPPMs leveraging hardwired models for user mobility. We consider *optimal* LPPM designs, i.e., defense mechanisms that, under a certain mobility model, maximize the privacy metric $\mathbf{P}_{\mathtt{AE}}$ against the best possible attack given a constraint on the maximum average loss $\overline{\mathrm{Q}}$ allowed. We note that, given a mobility model, there is a *familiy* of LPPMs that are *all optimal* (i.e., all of them achieve the maximum $\mathbf{P}_{\mathtt{AE}}$ given a constraint on $\overline{\mathrm{Q}}$), as proven in [92]. However, there are no universally optimal LPPMs *in practice*, i.e., when evaluated with testing data against an optimal attack. Thus, it is important to keep in mind that, even if two members of a family of optimal LPPMs perform equally in theory, they might perform differently in practice.

We explain LPPM design for the r-th release: the user is at location x^r and wants to query the LBS by releasing an obfuscated location z^r. The user knows all her previous real and obfuscated locations, i.e., $\mathbf{x}^{r-1}$ and $\mathbf{z}^{r-1}$, and the LBS/adversary knows the previously released locations $\mathbf{z}^{r-1}$. The optimal LPPM design problem can be written mathematically as

$$\begin{aligned} f = \underset{f}{\operatorname{argmax}} \quad & \min_{h} \mathbf{P}_{\mathtt{AE}}(f, h, r, r)\,, \\ \text{subject to} \quad & \overline{\mathrm{Q}}(f, r) \leq \overline{\mathrm{Q}}_{\mathtt{max}}. \end{aligned} \tag{5.10}$$

Note that f must satisfy some additional constraints since it is a probability density function, but we have omitted those from (5.10) for simplicity. Also, we have considered just the case where the user wants to protect her current location at time r. We note however that the user could set other goals, like trying to protect the privacy of future location releases $\mathbf{P}_{\mathtt{AE}}(f, h, r, s)$ for $s > r$, past locations ($s < r$), or a combination of both. Our findings could be adapted to such cases, but we do not study them here for simplicity and space restrictions.

We also limit ourselves to optimal output-based LPPMs, i.e., defenses that can be characterized by $f(z^r|\mathbf{z}^{r-1}, x^r)$ and do not depend on previous inputs $\mathbf{x}^{r-1}$. We do this to avoid the computational issues that stem from the fact that, in order to guarantee that an LPPM is optimal, the user has to assess its privacy against

an optimal attack. In order to do this with a full-LPPM $f(z^r|\mathbf{z}^{r-1}, \mathbf{x}^r)$, she has to characterize the posterior probability of the secret locations after releasing the obfuscated locations, i.e., $p(\mathbf{x}^r|\mathbf{z}^{r-1})$. If $x \in \mathcal{X}$ and $\mathcal{X}$ is discrete, this requires handling $|\mathcal{X}|^r$ values, which quickly becomes unfeasible for any computer (e.g., in a small map with $|\mathcal{X}| = 200$ discrete locations, if we represent a float with 4 bytes, to protect only $r = 8$ locations we would need over 1 million Terabytes). Since measuring the privacy against an optimal adversary is unfeasible in full-type LPPMs, we do not consider them in our design approaches.

Note that this computational issue is not a problem in output-based LPPMs. This is because, in this case, to assess the performance against an optimal adversary the user internally computes $p(x^r|\mathbf{z}^{r-1})$. She only needs to handle $|\mathcal{X}|$ parameters for this, since $\mathbf{z}^{r-1}$ have been seen in the past by both the user and the adversary, so they can be treated as fixed parameters at time r.

Below, we explain how to compute optimal LPPMs in the sporadic and Markov hardwired models.

5.4.1. LPPM Design in the Hardwired Sporadic Model

In the literature, we find many works that study LPPM design under the sporadic hardwired model for user mobility. Most works consider that the LPPM belongs to the memoryless type $f(z^r|x^r)$, either for tractability [46,92] or because they focus on single queries [65,67]. In Appendix 5.A, we formally prove that, in the hardwired model, a properly designed LPPM of the memoryless type *does not provide less privacy* than an LPPM of the full type $f(z^r|\mathbf{z}^{r-1}, \mathbf{x}^r)$. This means that considering full-type or output-based LPPMs just complicates the problem and does not provide any advantage over memoryless LPPMs.

There are two main approaches to compute optimal LPPMs in sporadic models:

Linear Programming Approaches. Shokri et al. provide a technique to design optimal LPPMs given any pair of functions $d_P(\cdot)$ and $d_Q(\cdot)$ [46]. This approach consists on solving a linear program, which can only be done, for computational reasons, if the spaces of real ($\mathcal{X}$) and obfuscated ($\mathcal{Z}$) locations are discrete. The program receives the mobility profile π which determines the distribution of x^r, and returns an optimal LPPM $f(z^r|x^r)$. If the number of discrete locations is N, the linear program contains $N(N+1)$ bounded variables, N^2+1 inequality constraints, and N equality constraints. Therefore, finding an optimal obfuscation mechanism using linear programming is only feasible if the number of discrete locations is modest.

Also, Oya et al. showed in [92] that the algorithm used to solve the linear program greatly affects the performance of the resulting LPPM in terms of other

privacy metrics (e.g., the conditional entropy). The recommendation in [92] is to use an interior-point algorithm, rather than a simplex algorithm.

Remapping Techniques. In [67], Chatzikokolakis et al. propose a technique called *optimal remapping* that provides an average loss improvement for any memoryless LPPM, without reducing privacy. They proposed this method under the hardwired sporadic mobility model. We used this technique in the previous chapter, but we summarize it here briefly for clarity, since we use it below. Let $\tilde{f}$ be an obfuscation mechanism, and let $\tilde{z}^r$ be an obfuscated location generated from x^r using such LPPM. Before reporting $\tilde{z}^r$, the user can compute the posterior $p(x^r|\tilde{z}^r)$ using $\pi(x^r)$ and $\tilde{f}$. With this posterior, she can compute an alternative obfuscated location z^r:

$$z^r = \underset{z^r}{\operatorname{argmin}} \sum_{x^r \in \mathcal{X}} p(x^r|\tilde{z}^r) \cdot d_Q(x^r, z^r)\,. \tag{5.11}$$

By reporting z^r (instead of $\tilde{z}^r$), the user achieves a reduction on her average loss (if the mobility profile π of the sporadic model used to compute (5.11) is close to her real behavior). Also, note that no information about the previous or current input is used in the remapping (since the posterior is computed only using the current output and π, which are known to the adversary). This means that, by performing this "remapping" from $\tilde{z}^r$ to z^r, the privacy of the resulting LPPM cannot decrease.

Later, in [92], Oya et al. proved that if the distance functions used to measure privacy and utility are the same (i.e., $d_P(\cdot) \equiv d_Q(\cdot)$), the LPPM that results from remapping *any* LPPM is optimal in the hardwired sporadic mobility model. This technique can even be applied to design LPPMs when their output space is the plane $\mathcal{Z} \equiv \mathbb{R}^2$. Overall, solving (5.11) is much faster than solving the linear program mentioned above, although it only yields optimal LPPMs if $d_P(\cdot) \equiv d_Q(\cdot)$.

5.4.2. LPPM Design in the Hardwired Markov Model

In the Markov model, the input locations $x^1, x^2, \ldots$ are correlated. This creates dependencies between past released locations $\mathbf{z}^{r-1}$ and the current location x^r, that the user must take into account when designing the LPPM.

To the best of our knowledge, the only approach to compute optimal LPPMs under the Markov mobility model consists on solving a linear program [91, 93]. We explain this approach, and then extend the remapping techniques of sporadic models so that we can efficiently design optimal LPPMs under the Markov model.

Linear Programming Approaches. Theodorakopoulos et al. [93] extend the linear programming approach of [46] to the non-sporadic location privacy case.

They propose a framework where the user can specify which obfuscated location(s) she wants to generate at time r, which real locations she wants to protect, and which obfuscated locations were released to the LBS in the past. In their implementation, they specifically consider a Markov model for user mobility. In the case we are studying, where the user wants to release z^r to protect x^r and $\mathbf{z}^{r-1}$ have already been released, the approach works as follows.

For the first release ($r = 1$), the user just takes the initial profile $\pi_0(x^1)$ and solves a linear program analogous to the sporadic location privacy one [46]. This produces an LPPM $f(z^1|x^1)$ that maximizes the privacy metric given a quality loss constraint. Then, she computes the posterior $p(x^1|z^1)$ using $\pi_0(x^1)$ and Bayes' formula, and uses it to obtain the probability distribution of the next real location given the released location: $p(x^2|z^1) = \sum_{x^1 \in \mathcal{X}} M(x^2|x^1) \cdot p(x^1|z^1)$.

For the next releases ($r > 1$), the steps are analogous, but they use $p(x^r|\mathbf{z}^{r-1})$ instead of π_0. Particularly, before the r-th query the user knows $p(x^r|\mathbf{z}^{r-1})$. With this probability distribution, the user can solve a linear program to find an optimal LPPM $f(z^r|\mathbf{z}^{r-1}, x^r)$. Then, she can compute the posterior using Bayes' formula:

$$p(x^r|\mathbf{z}^r) = \frac{f(z^r|\mathbf{z}^{r-1}, x^r) \cdot p(x^r|\mathbf{z}^{r-1})}{\sum_{\tilde{x}^r \in \mathcal{X}} f(z^r|\mathbf{z}^{r-1}, \tilde{x}^r) \cdot p(\tilde{x}^r|\mathbf{z}^{r-1})}\,, \tag{5.12}$$

and update it for the next step using the Markov transition matrix:

$$p(x^{r+1}|\mathbf{z}^r) = \sum_{x^r \in \mathcal{X}} M(x^{r+1}|x^r) \cdot p(x^r|\mathbf{z}^r)\,. \tag{5.13}$$

In [91,93], the authors evaluate their LPPMs theoretically, i.e., they compute the average adversary error and average loss that the user would have if she followed the Markov model using the analytical expressions (5.3) and (5.6). For example, they compute $f(z^2|z^1, x^2)$ for all possible values of z^2, z^1, x^2. Therefore, for computational reasons, they do not evaluate the performance of these LPPMs for more than $r = 3$ consecutive locations. During an empirical evaluation, however, one does not need to store all possible values of these variables. Since the past obfuscated locations $\mathbf{z}^{r-1}$ are known both to the user and the adversary, the user can just compute $f(z^r|\mathbf{z}^{r-1}, x^r)$ by assuming that $\mathbf{z}^{r-1}$ is fixed. Therefore, the computational cost of computing this Markov-based LPPM in each query is the same as solving the linear program in the sporadic case.

Remapping Techniques. Even though the complexity of the linear programming approach in the Markov scenario is the same as in the sporadic scenario, if the number of discrete locations we consider is not small, finding an optimal LPPM is still computationally expensive. To solve this issue, we extend the remapping techniques to the Markov scenario. To the best of our knowledge, this is the first time these techniques are extended beyond the sporadic location privacy scenario.

Assume that, at the time of the r-th location release, the user has computed $p(x^r|\mathbf{z}^{r-1})$ according to the Markov model. Let $\tilde{f}$ be any memoryless-LPPM $\tilde{f}(\tilde{z}^r|x^r)$. The user uses this LPPM to generate a temporary $\tilde{z}^r$, and then computes the posterior

$$p(x^r|\tilde{z}^r, \mathbf{z}^{r-1}) = \frac{\tilde{f}(\tilde{z}^r|x^r) \cdot p(x^r|\mathbf{z}^{r-1})}{\sum_{\tilde{x}^r \in \mathcal{X}} \tilde{f}(\tilde{z}^r|\tilde{x}^r) \cdot p(\tilde{x}^r|\mathbf{z}^{r-1})} \,. \tag{5.14}$$

With this posterior, she can then compute the final location that she releases

$$z^r = \operatorname*{argmin}_{z^r} \sum_{x^r \in \mathcal{X}} p(x^r|\tilde{z}^r, \mathbf{z}^{r-1}) \cdot d_Q(x^r, z^r) \,. \tag{5.15}$$

This process defines a new LPPM $f(z^r|\mathbf{z}^{r-1}, x^r)$. At this point, the user can compute $p(x^{r+1}|\mathbf{z}^r)$ for the next release following (5.12) and (5.13). Computing the LPPM by solving (5.15) is much faster than solving the linear program explained above. Also, the LPPM that results form the remapping can be shown to be optimal in the hardwired Markov model if $d_P(\cdot) \equiv d_Q(\cdot)$ (c.f. [92]).

5.5. Evaluation: Optimal Hardwired LPPMs

In this section, we evaluate the optimal LPPMs developed for hardwired models that we described in Sect. 5.4 using the evaluation framework described in Section 5.2.2. For readability and clarity, we use the term `SP-LPPM` to denote a generic LPPM that is optimal under the hardwired **SP**oradic mobility model [46,67,92]. This LPPM can be computed by following any of the techniques explained in Sect. 5.4.1. Likewise, we use `MK-LPPM` to denote an LPPM that is optimal under the hardwired **M**ar**K**ov mobility model [91,93] (we can compute it as explained in Sect. 5.4.2). Note that `SP-LPPM` and `MK-LPPM` define *families* of optimal LPPMs (i.e., there are infinite instantiations of them that meet their optimality conditions).

We perform two different experiments: one to evaluate `SP-LPPM` in the sporadic location release scenario (Experiment SP), and another one to evaluate `MK-LPPM` in the continuous location release (Experiment MK). For these experiments, we consider three datasets, two different instantiations of `SP-LPPM` and `MK-LPPM`, and two optimal attacks. We explain these choices below. Table 5.2 summarizes the new terminology of this evaluation, and Table 5.3 shows the configuration of our experiments.

Datasets. We consider three datasets: Brightkite[1], Gowalla[2], and TaxiCab

[1] `https://snap.stanford.edu/data/loc-brightkite.html`
[2] `https://snap.stanford.edu/data/loc-gowalla.html`

Table 5.2: Terminology for the experiments.

`SP-LPPM`	Family of optimal LPPMs developed with the hardwired *sporadic* mobility mode (Sect. 5.4.1).
`MK-LPPM`	Family of optimal LPPMs developed with the hardwired *Markov* mobility mode (Sect. 5.4.2).
`SP-LH`	Optimal location hiding LPPM from the `SP-LPPM` family.
`SP-Exp`	Optimal exponential LPPM from the `SP-LPPM` family.
`MK-LH`	Optimal location hiding LPPM from the `MK-LPPM` family.
`MK-Exp`	Optimal exponential LPPM from the `MK-LPPM` family.

Table 5.3: Summary of the experiments to evaluate hardwired LPPMs.

	Experiment SP `SP-LPPM`	**Experiment MK** `MK-LPPM`
Evaluation target		
Datasets	Gowalla (shuffled) Brightkite (shuffled)	Gowalla Brightkite TaxiCab
Distance function	Manhattan ($d_P \equiv d_Q$)	
LPPM	Loc. Hiding (`SP-LH`) Exponential (`SP-Exp`)	Loc. Hiding (`MK-LH`) Exponential (`MK-Exp`)
Attack we evaluate	Optimal Sporadic	Optimal Markov

traces from CRAWDAD.[3] Each dataset contains location traces identified by the user ID, latitude, longitude, and timestamp. We take user check-ins inside the San Francisco region (we take the region between latitude coordinates 37.5500 and 37.8010, and longitude coordinates -122.5153 and -122.3789). Then, we quantize the area into 25×10 regions and consider the centers of those regions as our alphabets $\mathcal{X} = \mathcal{Z} = \hat{\mathcal{X}}$, as in [91, 93].

Gowalla and Brightkite are examples of datasets with very sparse check-in behavior (e.g., in Gowalla, each user has an average of 60 check-ins during over 20 months of data collection). Thus, in these datasets we separate 20 users that have at least 300 check-ins inside the San Francisco region, regardless of when those check-ins were made, and save the remaining check-ins of all the other users together (around 35 000 in Brightkite and 75 600 in Gowalla).

Regarding the training/testing separation, in our experiments, we evaluate the performance of the last 5 users in these datasets. We consider two training settings: in the first setting, that we call *scarce training*, the users train their LPPMs with the traces of the first 15 users (4 500 locations). In the second setting, that we call *rich training*, each user trains her model using the check-ins of all the other users in the dataset (35 000 in Brightkite, 75 600 in Gowalla). This is depicted in Fig. 5.5a.

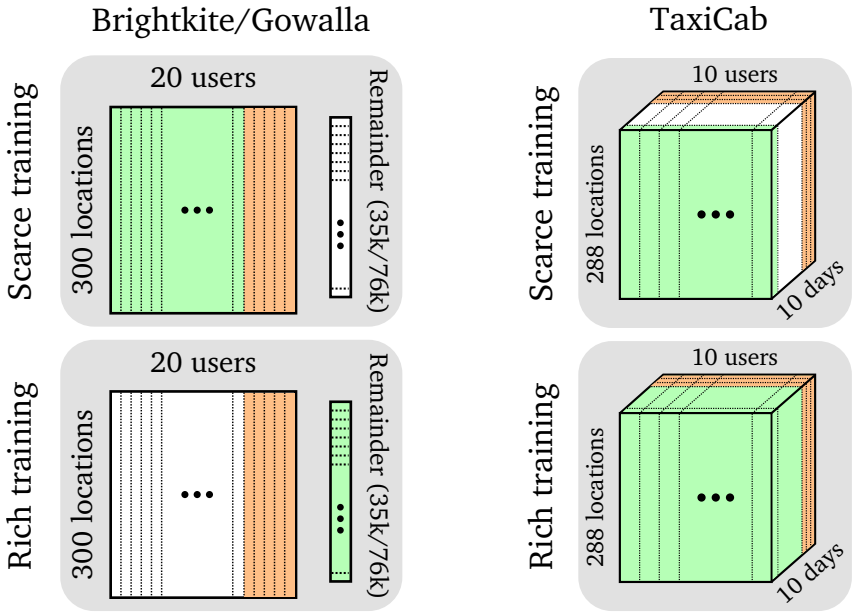
TaxiCab contains very dense location reports of cabs in the San Francisco region over 30 days. In this case, we organize each user's traces by days, and discard those days where the user remains silent for more than 2 hours. Then, we select 10 users for which we retain at least 10 days. For each trace, we select one check-in for each period of 5 minutes (considering that the user remained in the same location if she did not perform a new check-in in the last 5 minutes). This way, we build, for each user, a set of 10 days with 288 check-ins ($288 \cdot 5$ minutes $= 1$ day).

In this dataset, we evaluate the performance of each user in her last 3 days. We consider two settings for the training data: in our first setting (scarce), each user uses her first day as training data. In our second setting (rich), each user trains her model using her first 7 days of data (Fig. 5.5b).

Training the LPPMs. We explain how the users estimate the parameters of the models that they use to build optimal LPPMs. For the LPPMs built using the hardwired sporadic model (`SP-LPPM`), each user computes π as a normalized histogram of the training set traces, i.e., she counts the number of check-ins in each location $x \in \mathcal{X}$ in the training data and normalizes by the total number of check-ins.

For LPPMs built using the hardwired Markov model (`MK-LPPM`), each user computes π_0 as a normalized histogram of the location check-ins in the training

[3] `https://crawdad.org/epfl/mobility/20090224/`



(a) Brightkite/Gowalla datasets.

(b) Taxicab dataset.

Figure 5.5: Processed datasets that we collected. We consider two training settings for each dataset (scarce and rich). For each of these settings, the figure displays the training data in green, and the testing data in orange.

data, and builds the transition probabilities $M(x_i|x_j)$ by counting the number of transitions from x_j to x_i in the data, and normalizing.

LPPM instantiations. For each family of optimal LPPMs (`SP-LPPM` and `MK-LPPM`) we test the performance of two different instantiations:

- `LH` refers to *location hiding*: for each input location x^r, the user chooses randomly between revealing her real location $z^r = x^r$ (with probability α) or not revealing any information (with probability $1 - \alpha$). We model the second case as picking uniformly at random another location of the map. We test 10 values of $\alpha = 0, 0.1, 0.2, \ldots, 1$ to study the trade-off between $\mathbf{P}_{\texttt{AE}}$ and $\overline{\text{Q}}$. We apply an optimal remapping to this LPPM to make it optimal: the remapping in Sect. 5.4.1 gives us an `SP-LPPM` that we denote `SP-LH`, and the remapping in Sect. 5.4.2 gives us an `MK-LPPM` that we denote `MK-LH`.

- `Exp` is the *exponential* LPPM [69]: this LPPM reports location z^r with a probability proportional to $\exp(-d_Q(x^r, z^r) \cdot \epsilon)$ (i.e., it has an exponentially decreasing probability of reporting locations that are far from the real location). We test 10 values of $\epsilon = 0\text{km}^{-1}, 0.02\text{km}^{-1}, 0.04\text{km}^{-1}, \ldots, 0.02\text{km}^{-1}$ to tune the average loss and privacy of this LPPM. We apply an optimal remapping to this LPPM to build an optimal defense (denoted `SP-Exp` and `MK-Exp` for the sporadic and Markov models, respectively).

Attacks. As we mentioned in Section 5.2.2, we consider a worst-case adversary which deploys optimal attacks constructed with information about the testing data. The optimal attacks, after observing z^r, compute the posterior $p(x^r|\mathbf{z}^r)$ and pick the $\hat{x}^r$ that minimizes the privacy $\mathbf{P}_{\texttt{AE}}$. We consider two attacks: a sporadic-based and a Markov-based attack. These attacks use the actual mobility profiles and transition matrices of the users (i.e., computed from the testing data) to perform their estimation $\hat{x}^r$.

In all of our experiments, we use the Manhattan distance as the distance metric for privacy $d_P(\cdot)$ and utility $d_Q(\cdot)$. We think this is a reasonable choice, since our traces belong to metropolitan areas, where the Manhattan distance between two points is close to the physical distance that a car/person has to traverse to move from one point to the other. We measure distance in kilometers (km), but this could be converted to time (by dividing it by speed) or another metric related to the physical distance between two points.

We note that, since we chose $d_P(\cdot) \equiv d_Q(\cdot)$, the theoretical performance of any optimal LPPM is $\mathbf{P}_{\texttt{AE}} = \overline{\mathrm{Q}}$, as shown empirically in [46] and proven analytically in [92]. This means that any optimal LPPM evaluated in the same data used for its training would achieve $\mathbf{P}_{\texttt{AE}} = \overline{\mathrm{Q}}$. This is true for `SP-LH` and `SP-Exp` against the optimal sporadic attack, and for `MK-LH` and `MK-Exp` against the optimal Markov attack. We see below that, when these optimal LPPMs are evaluated on a testing set that is different from the training data, they do not achieve this optimal privacy level, i.e., in practice, $\mathbf{P}_{\texttt{AE}} < \overline{\mathrm{Q}}$.

We explain how we generate the plots in our evaluation. Given a particular experiment, user, and LPPM setting, we compute $\overline{\mathrm{Q}}(f, r)$ and $\mathbf{P}_{\texttt{AE}}(f, h, r, r)$ by averaging 100 repetitions of our experiment (i.e., we repeat the process of computing the LPPM, generating obfuscated locations and computing the adversary estimation 100 times). Then, we average the performance over r (i.e., $\overline{\mathrm{Q}} \doteq 1/\rho \sum_{r=1}^{\rho} \overline{\mathrm{Q}}(f, r)$ and $\mathbf{P}_{\texttt{AE}} \doteq 1/\rho \sum_{r=1}^{\rho} \mathbf{P}_{\texttt{AE}}(f, h, r, r)$). This gives us, for each user that we evaluate, points along their $\mathbf{P}_{\texttt{AE}}$ vs. $\overline{\mathrm{Q}}$ performance line. Finally, we generate quality loss values linearly spaced between 0 and 4km and, using linear interpolation, compute the average, maximum and minimum privacy over the users for each of those quality loss values. All of our experiments are conducted using Python 3.

5.5.1. Experiment SP: Sporadic Hardwired LPPMs

We evaluate `SP-LH` and `SP-Exp` against the optimal sporadic-based attack that uses the real mobility profile of the user. We use only Gowalla and Brightkite datasets for this experiment, since Taxicab is more characteristic of non-sporadic mobility behaviors. For each simulation of this experiment, we randomly shuffle the user traces (i.e., each column in the matrix represented in Fig. 5.5a). We do

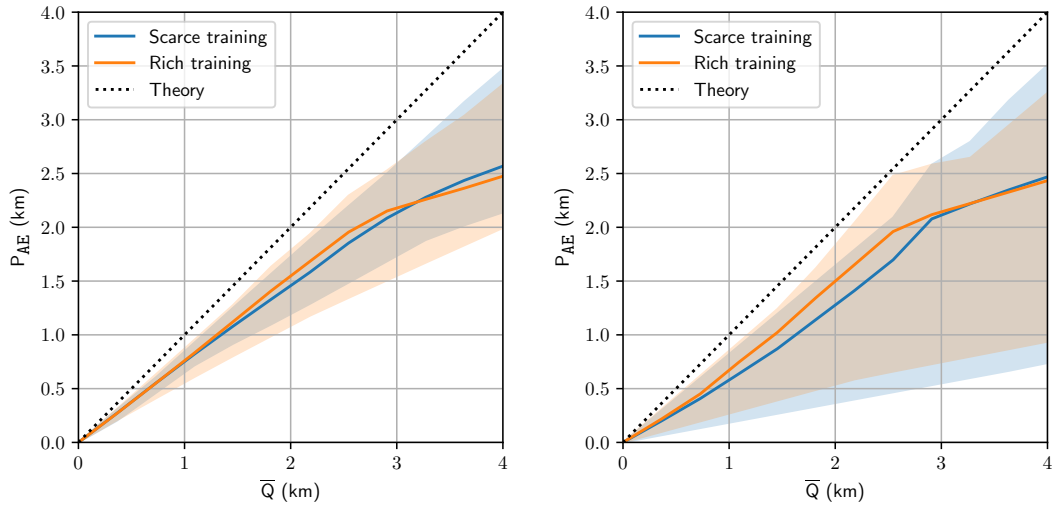
this to break any possible timing correlation that remains in these datasets and ensure that our evaluation of these LPPMs is fair.

Figure 5.6 shows the results, where the blue and orange lines represent the average privacy of the users when they use the scarce and rich training data, respectively. The shaded area represents the minimum and maximum privacy among the users that we evaluate. We see that, in both datasets, and regardless of the LPPM type, the privacy of the users evaluated with a testing data that differs from the training information is below the theoretical value $\mathbf{P}_{\texttt{AE}} = \overline{\text{Q}}$. Also, training with the rich training set provides more privacy on average, since this dataset has more information about the sporadic check-in behavior of the users ($35\,000 - 75\,600$ check-ins, versus $4\,500$ check-ins of the scarce dataset). However, this improvement is slight: none of the training sets capture the real user behavior precisely, since both contain data from different users. Some of the users that we evaluate have a behavior that is particularly different from the training data (e.g., lower shaded area in Fig. 5.6b), and thus achieve very low privacy. This experiment shows that training an optimal sporadic LPPM with location data from other users (e.g., [67]) is very dangerous from a privacy standpoint.

5.5.2. Experiment MK: Markov Hardwired LPPMs

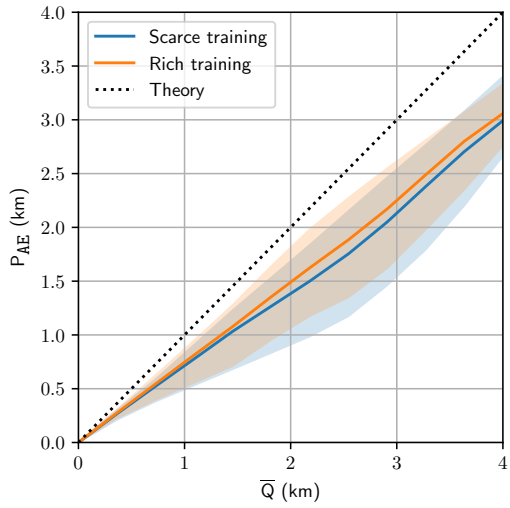
We evaluate `MK-LH` and `MK-Exp` against the optimal Markov adversary. Figure 5.7 shows the performance in Brightkite and Gowalla, and Fig. 5.8 shows the performance in TaxiCab dataset. The results in Brightkite and Gowalla are very similar to the ones in the previous experiment, i.e., an optimal Markov LPPM that has been designed by hardwiring it on training data from other users provides significantly less privacy than expected in theory.

The results in TaxiCab dataset, however, are significantly better for the users. This is because, in this dataset, we have continuous location data (i.e., one location reported every 5 minutes). This means that two consecutive locations are highly correlated because of road restrictions (e.g., one-way roads, mandatory turns, etc.). Cabs follow very different paths each day, and thus it would seem that training their LPPMs with past data should not be significantly beneficial for them. However, the training data encodes these road restrictions. This is very important: the optimal Markov LPPMs are thus designed taking these constraints into account. Since the road restrictions are also part of the testing data, the optimal Markov LPPM is able to get close to optimal performance during evaluation. We also observe that training with seven days of data (rich training) is slightly better than training with a single day (scarce training). This slight improvement suggests that a single day of training already encodes most of the road restrictions. To validate this hypothesis, we also conducted experiments where we train each user's LPPM with past location traces of different users, and the results were similar (c.f. [97]). This confirms that taking road constraints

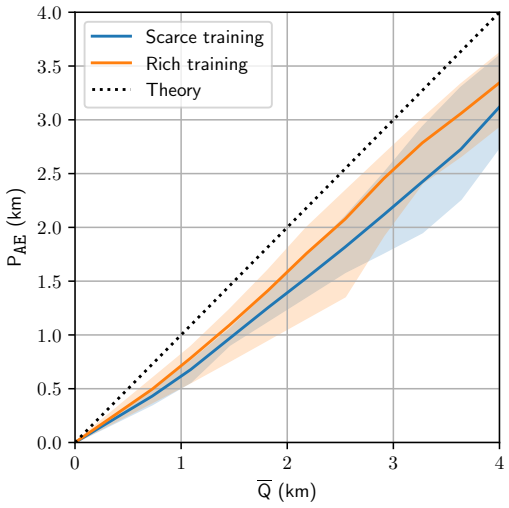


(a) Brightkite, `SP-Exp`.

(b) Brightkite, `SP-LH`.



(c) Gowalla, `SP-Exp`.



(d) Gowalla, `SP-LH`.

Figure 5.6: Experiment SP. Performance of `SP-Exp` and `SP-LH` against the optimal sporadic attack in Brightkite and Gowalla datasets (with shuffled traces).

into account is of paramount importance towards achieving high protection in continuous non-sporadic location privacy.

Finally, `MK-Exp` performs better than `MK-LH`. These results support the findings in [92], where authors showed that exponential mechanisms perform significantly better than location hiding techniques when evaluated with metrics they have not been optimized for. In our case, we see that `MK-Exp` performs better than `MK-LH` when evaluated in testing data it has not been optimized for.

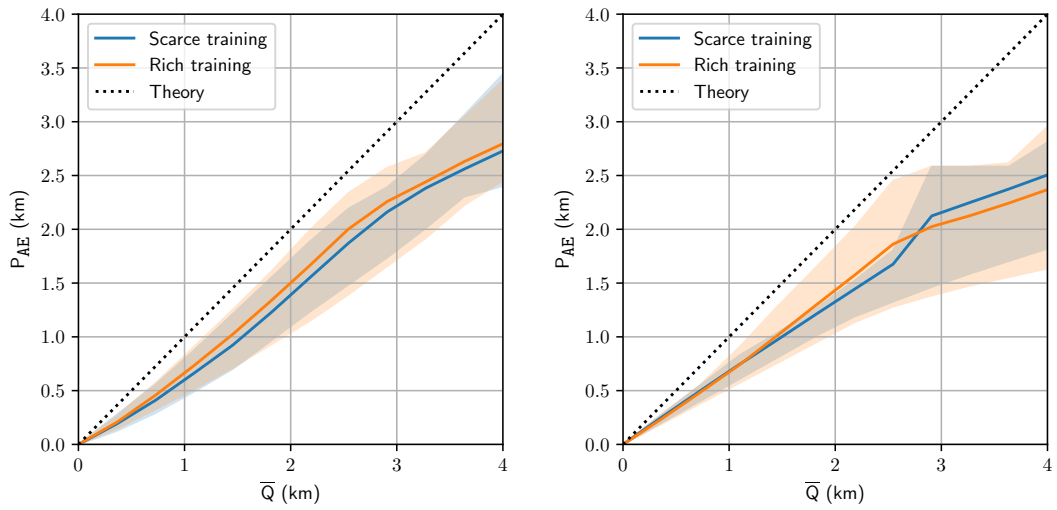
5.6. Blank-Slate Models

We have seen that hardwiring the training data into the mobility models used for LPPM design can be detrimental to privacy. To alleviate this issue, we propose *blank-slate* models for user mobility. These models treat their parameters (π in the sporadic case; or π_0 and M in the Markov case) as unknown variables that are never completely known to the user when designing her LPPM. These parameters can be initialized a-priori with training data, but do not remain fixed. Instead, the user updates them a-posteriori, as she acquires additional information from the observations (e.g., $\mathbf{x}$ and $\mathbf{z}$, from the testing set). Therefore, we can expect that LPPMs developed with blank-slate models will be desirable in situations where the training data does not adequately capture the user's mobility traits, either because it does not contain sufficient information or because it captures mobility patterns that are not characteristic of the user in question.

There are many ways in which a user can implement a blank-slate model. For example, a user can train a distribution on the hidden parameter (e.g., $p(\pi)$) based on training data, and then estimate this parameter a-posteriori using $\mathbf{x}$ and $\mathbf{z}$ (e.g., a maximum a-posteriori approach). In our case, we take a maximum likelihood approach that we explain below. We present a new family of LPPMs, the Profile Estimation-Based LPPMs (PEB-LPPMs), that we build by leveraging a *sporadic blank-slate model* for user mobility. We do not tackle the problem of LPPM design under blank-slate Markov mobility models, but we show that our PEB-LPPM is also useful for users whose mobility model is Markovian.

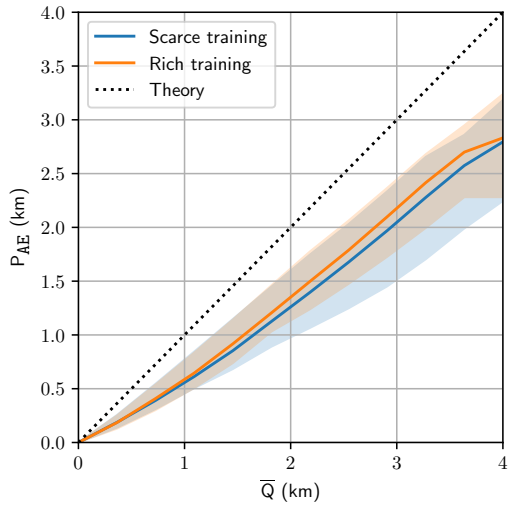
5.6.1. LPPM Design in the Sporadic Blank-Slate Model

A sporadic blank-slate model is characterized by a mobility profile π that is unknown to the user. In order to design an LPPM using this model, the user must first estimate this mobility profile. We propose to use a Maximum Likelihood Estimator (MLE) of the mobility profile before each query r, and then use this profile to build an optimal sporadic LPPM. We call the LPPM designed this way Profile Estimation Based (PEB)-LPPM.

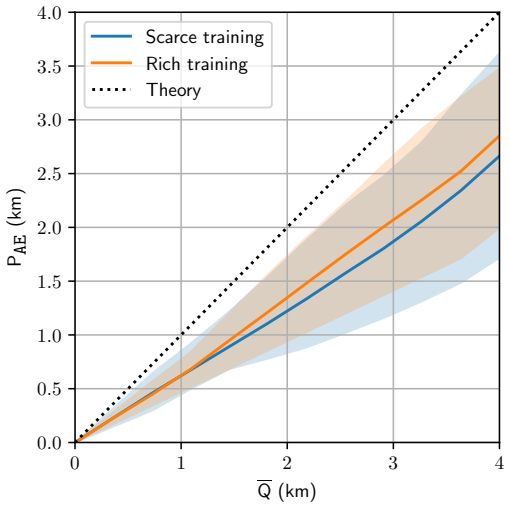


(a) Brightkite, `MK-Exp`.

(b) Brightkite, `MK-LH`.

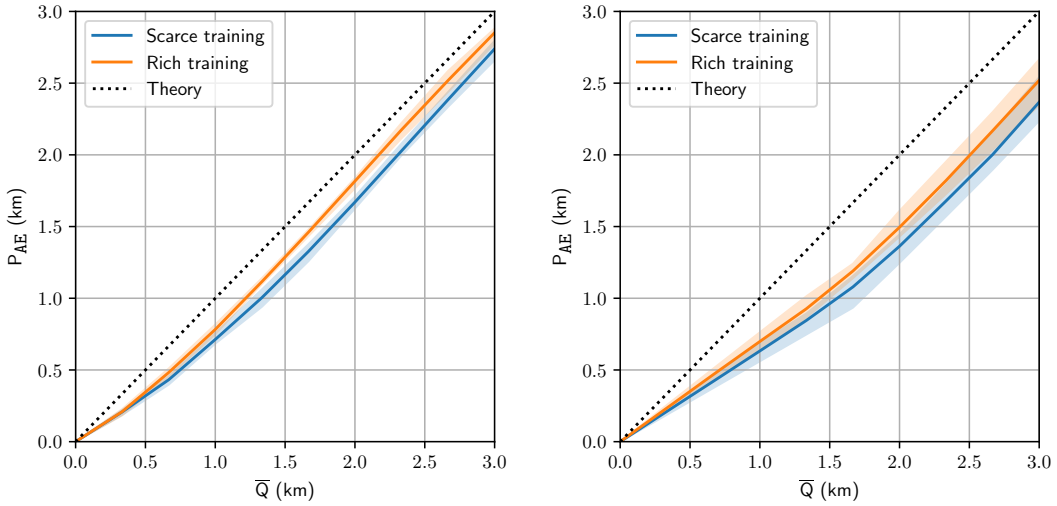


(c) Gowalla, `MK-Exp`.



(d) Gowalla, `MK-LH`.

Figure 5.7: Experiment MK: Performance of `MK-Exp` and `MK-LH`, against the optimal Markov attack in Brightkite and Gowalla datasets.



(a) TaxiCab, `MK-Exp`.

(b) TaxiCab, `MK-LH`.

Figure 5.8: Experiment MK: Performance of `MK-Exp` and `MK-LH` against the optimal Markov attack in Taxicab dataset.

More precisely, a PEB-LPPM is an output-based defense, i.e., characterized by $f(z^r|\mathbf{z}^{r-1}, x^r)$, that is computed by following these steps:

1. Compute an MLE of the mobility profile π using $\mathbf{z}^{r-1}$. Let this estimate be $\hat{\pi}^r_{ML}$.

2. Normalize the estimate $\hat{\pi}^r_{ML}$ to avoid variance issues for low r, producing $\hat{\pi}^r$.

3. Compute the optimal LPPM in the sporadic mobility model using $\hat{\pi}^r$ and generate z^r randomly using x^r and this newly created LPPM.

This whole process can be embedded into a function of the form $f(z^r|\mathbf{z}^{r-1}, x^r)$ that defines the PEB-LPPM.

Note that, in the first step above, the user could have also used her past real locations $\mathbf{x}^{r-1}$ to compute the MLE estimation of her mobility profile (since she knows them). This, effectively, would result on a full-LPPM $f(z^r|\mathbf{z}^{r-1}, \mathbf{x}^r)$. As we mentioned in Section 5.4, assessing the privacy of these LPPMs against an optimal adversary is computationally intractable. Thus, to avoid gaps in our evaluation, we only use $\mathbf{z}^{r-1}$ to compute our MLE of the mobility profile. We delve into the steps of the PEB-LPPM design below.

5.6.2. Step 1: Mobility Profile Estimation.

We derive the Maximum Likelihood Estimator (MLE) of the mobility profile given $\mathbf{z}^r$. We use $\pi_i \equiv p(x = x_i)$ to denote the probability mass function defined by π. Let $\mathcal{P}$ be the set of all the possible mobility profiles, i.e., $\mathcal{P} \doteq \{\pi | \sum_{i=1}^{|\mathcal{X}|} \pi_i = 1, \quad \pi_i \geq 0\}$. The MLE of π given $\mathbf{z}^r$ is defined as

$$\hat{\pi}^r_{ML} = \underset{\pi \in \mathcal{P}}{\operatorname{argmax}}\, p(\mathbf{z}^r|\pi)\,. \tag{5.16}$$

An efficient iterative way of computing this estimator is the Expectation-Maximization (EM) method [98]. Instead of maximizing $p(\mathbf{z}^r|\pi)$, we rely on $\mathbf{x}^r$ as auxiliary data and define a Q function as

$$Q(\pi, \pi^t) = \mathrm{E}\left\{\log p(\mathbf{x}^r|\pi)|\mathbf{Z} = \mathbf{z}^r, \Pi = \pi^t\right\}. \tag{5.17}$$

The EM method iterates over two steps: first, compute $Q(\pi, \pi^t)$ (E-step), and then find π^{t+1} as the profile π that maximizes $Q(\pi, \pi^t)$ (M-step). We expand Q as

$$\begin{aligned} Q(\pi, \pi^t) &= \mathrm{E}\left\{\log p(\mathbf{x}^r|\pi)|\mathbf{Z}^r = \mathbf{z}^r, \Pi = \pi^t\right\} \\ &= \sum_{s=1}^{r} \mathrm{E}\left\{\log p(x^s|\pi)|\mathbf{Z}^r = \mathbf{z}^r, \Pi = \pi^t\right\} \\ &= \sum_{s=1}^{r}\sum_{i=1}^{|\mathcal{X}|} \log \pi_i \cdot p(x^r_i|\mathbf{z}^r, \pi^t) \\ &= \sum_{i=1}^{|\mathcal{X}|} \log \pi_i \cdot \left[\sum_{s=1}^{r} p(x^s_i|\mathbf{z}^r, \pi^t)\right]. \end{aligned} \tag{5.18}$$

In order to find the $\pi \in \mathcal{P}$ that maximizes $Q(\pi, \pi^t)$, we build the Lagrange multipliers function

$$L(\pi, \lambda, \boldsymbol{\mu}) = Q(\pi, \pi^t) + \lambda\left(\sum_{i=1}^{|\mathcal{X}|} \pi_i - 1\right) + \sum_{i=1}^{|\mathcal{X}|} \mu_i \pi_i\,, \tag{5.19}$$

where the term with λ corresponds to the constraint $\sum_{i=1}^{|\mathcal{X}|} \pi_i = 1$ and the terms with μ_i correspond to $\pi_i \geq 0$. We take $\mu_i = 0$ for the non-negativity constraints, and by solving $\partial L/\partial \pi_i = 0$ and $\partial L/\partial \lambda = 0$ we obtain the maximum, which gives us the update rule

$$\pi_i^{t+1} = \frac{1}{r}\sum_{s=1}^{r} p(x^s_i|\mathbf{z}^s, \pi^t) = \frac{1}{r}\sum_{s=1}^{r} \frac{\pi^t_i \cdot f(z^s|\mathbf{z}^{s-1}, x^s_i)}{\sum_{k=1}^{|\mathcal{X}|} \pi^t_k \cdot f(z^s|\mathbf{z}^{s-1}, x^s_k)}\,. \tag{5.20}$$

Following [99], we can see that this solution is the global maximum of $Q(\pi, \pi^t)$, since it meets the KKT (Karush-Kuhn-Tucker) conditions, $Q(\pi, \pi^t)$ is strictly concave on π (it is a weighted sum of logarithms) and $\mathcal{P}$ is a convex set.

Summarizing, in order to compute the MLE of the mobility profile, one proceeds as follows. First, define an initial profile π^0. Then, follow the update rule given by (5.20) until convergence (i.e., until the change from π^t to π^{t+1} is small enough). This algorithm is ensured to converge to the MLE for memoryless and output-based LPPMs, as we prove in Appendix 5.B.

5.6.3. Step 2: MLE Normalization

The accuracy of the MLE estimator above depends on the number of queries done previously. For example, we can expect to have a worse estimation of π if we compute it at time $r = 2$ using only z^1, compared to computing it at time $r = 100$ with $\mathbf{z}^{99}$. To alleviate this issue, we perform a normalization step. Let π_{ini} be a initial mobility profile (e.g., a uniform profile, a profile computed from auxiliary data, or a profile computed from the training data as in hardwired models) and $\gamma > 0$ be a constant. The final mobility profile after the normalization step is

$$\hat{\pi}^r = \frac{1}{r^\gamma} \cdot \pi_{ini} + \left(1 - \frac{1}{r^\gamma}\right) \cdot \hat{\pi}^r_{ML} \,. \tag{5.21}$$

The coefficient γ tunes how fast the effect of π_{ini} in $\hat{\pi}^r$ fades with r. For example, if the user does not have enough data to compute a reliable initial profile π_{ini}, she can simply set $\gamma = 0.5$ so that $\hat{\pi}^r$ converges fast to the ML estimation $\hat{\pi}^r_{ML}$. If the user believes that π_{ini} is representative of her current mobility behavior, a slower rate $\gamma = 0.1$ is more appropriate.

5.6.4. Step 3: Final LPPM Computation

Once the user has computed her estimation of the mobility profile $\hat{\pi}^r$ she builds an optimal memoryless LPPM for the sporadic location privacy case (e.g., using the linear programming or the optimal remapping approach we explained in Section 5.4.1). Using this LPPM, she samples the obfuscated location z^r given her real location x^r.

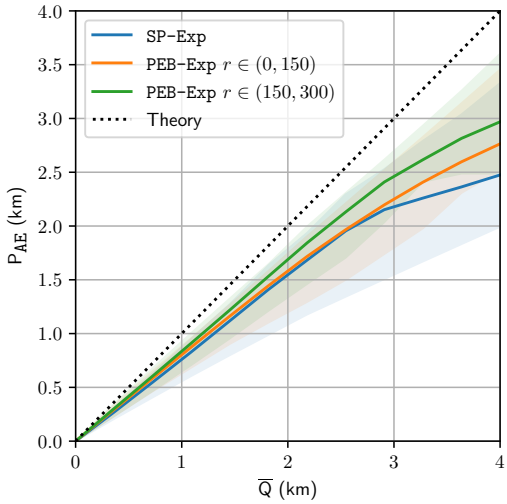
5.7. Evaluation of Profile Estimation-Based LPPMs

Now, we evaluate the performance of the PEB-LPPMs that we developed leveraging the blank-slate sporadic mobility model, and compare them with the optimal LPPMs that we evaluated earlier. We use the notation `PEB-LH` and `PEB-Exp` to denote the location hiding and exponential LPPMs computed following the PEB-LPPM strategy in Sect. 5.6. We heuristically chose to use the parameter $\gamma = 0.5$ in our experiments, so that the PEB-LPPMs adapt quickly to the MLE of the mobility profile. For example, this means that, after $r = 100$ queries, the mobility profile that is used for design $\hat{\pi}^r$ in (5.21) will be $\hat{\pi}^r = 0.1 \cdot \pi_{train} + 0.9 \cdot \hat{\pi}^r_{ML}$.

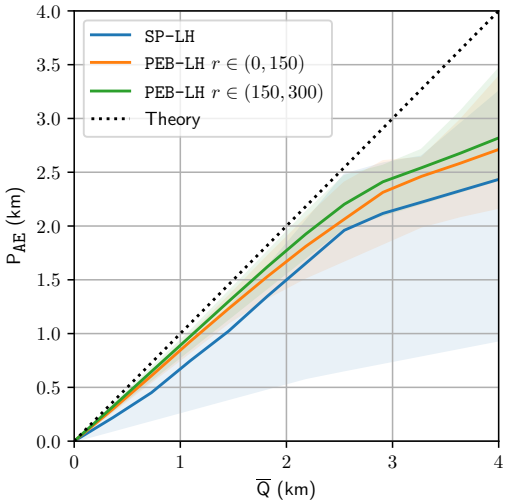
As in Section 5.5, we split the evaluation into two parts, using the same settings (see Table 5.3). Since PEB-LPPMs learn the user behavior as she queries the LBS, we can expect that their performance will improve over time. Therefore, instead of averaging $\overline{\mathrm{Q}}(f, r)$ and $\mathbf{P}_{\mathtt{AE}}(f, h, r, r)$ over all values of r, we perform the average over the first and last halves separately (e.g., $\sum_{r=1}^{150} \cdots$ and $\sum_{r=151}^{300} \cdots$ in Brightkite/Gowalla).

5.7.1. Experiment SP with PEB-LPPMs

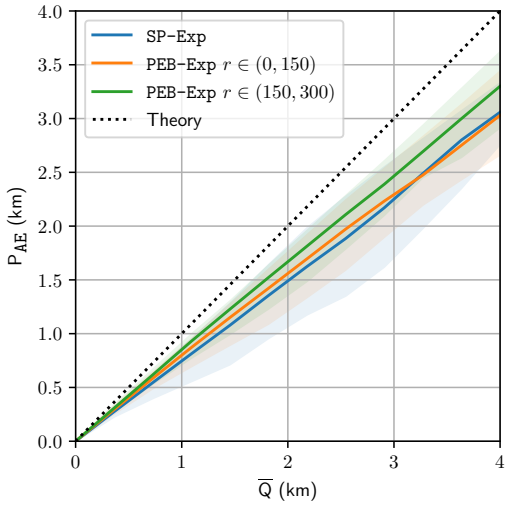
First, we evaluate PEB-LPPMs in the sporadic scenario. We compare the performance of `PEB-LH` and `PEB-Exp` with `SP-LH` and `SP-Exp`, against the optimal sporadic adversary. We use only the rich data to train `SP-LH` and `SP-Exp` and to initialize `PEB-LH` and `PEB-Exp` (for simplicity). Figure 5.9 shows the results. The blue line corresponds to the orange line in Fig. 5.6 (`SP-LPPM` trained with the rich data). The orange line is the average performance of PEB-LPPMs in the first 150 samples, and the green line is the average performance in last 150 samples. We can see that PEB-LPPMs always outperform hardwired ones (`SP-LPPM`) in the sporadic scenario, and that the performance of PEB-LPPMs improves with r. This is reasonable, as these mechanisms estimate the real user behavior adaptively during the evaluation, and with higher r values this estimation is more accurate. These results show that disregarding the training data and relying solely on the MLE of the mobility profile (`PEB-LH` and `PEB-Exp` with $r > 150$) can yield LPPMs that offer better protection than those hardwired on the training data (`SP-LH` and `SP-Exp`).



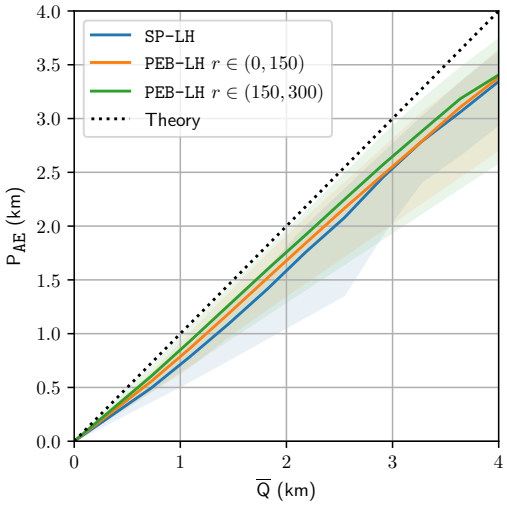
(a) Brightkite, `Exp`.



(b) Brightkite, `LH`.



(c) Gowalla, `Exp`.



(d) Gowalla, `LH`.

Figure 5.9: Experiment SP: Performance of PEB-LPPMs versus `SP-LPPM`, using Brightkite and Gowalla datasets (shuffled). `SP-LPPM` have been trained with the rich data.

5.7.2. Experiment MK with PEB-LPPMs

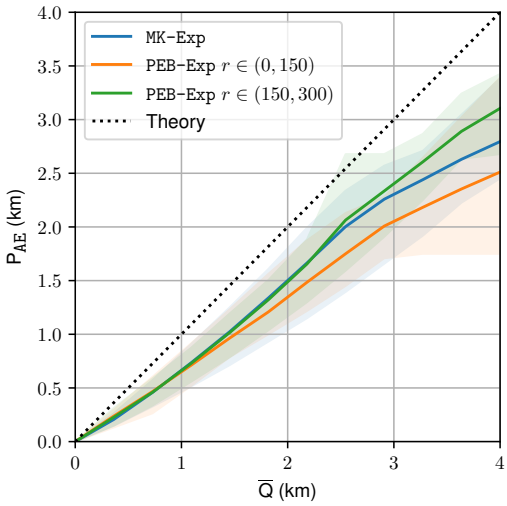
Now, we compare `PEB-LH` and `PEB-Exp` with `MK-LH` and `MK-Exp` against the optimal Markov adversary, in the settings of Experiment MK. Figure 5.10 shows the results for Brightkite and Gowalla, and Fig. 5.11 for TaxiCab. In Brightkite and Gowalla, we use only the rich training data to build `MK-LPPM`. Here, even though PEB-LPPMs are built upon the sporadic blank-slate mobility model, they are on-par with optimal Markov designs in non-sporadic location privacy settings, and in many cases outperform them. This is because Brightkite and Gowalla are datasets where user check-ins are not strongly correlated. This means that capturing the road restrictions is not decisive towards achieving a good privacy performance, and therefore PEB-LPPMs can compete with `MK-LPPM`.

The situation changes drastically in TaxiCab dataset (Fig. 5.11). In this case, even though we have decided to train `MK-LPPM` using the scarce training set (one day of data for each user), this is enough for `MK-LPPM` to achieve an outstanding performance (as we saw in Fig. 5.8). This is because, in TaxiCab dataset, the locations are tightly correlated due to road restrictions. PEB-LPPMs are built leveraging a sporadic blank-slate model, so they cannot capture these restrictions, and thus perform poorly in this dataset. Note that increasing r does not have a significant effect in the performance, since it does not matter how accurately the profile estimation of PEB-LPPMs is: a (sporadic) mobility profile cannot capture the correlations of non-sporadic models.

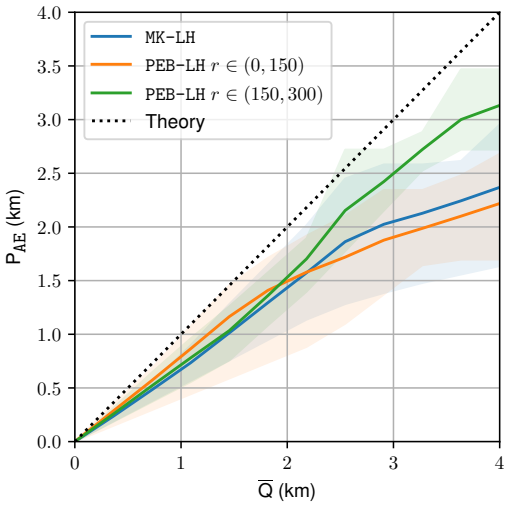
5.7.3. Summary of Results and Other Privacy Metrics

PEB-LPPMs outperform optimal hardwired LPPMs in all of our *sporadic* location privacy experiments. This is reasonable, as in these experiments the training data cannot closely characterize the behavior of the testing set users. This does not mean that PEB-LPPMs *always* outperform hardwired LPPMs in sporadic location release scenarios: if user behavior can be accurately modeled by the training data, the performance of hardwired LPPMs would be close to optimal. However, we can confirm that PEB-LPPMs are a powerful tool to protect users whose mobility behavior cannot be predicted from the training data.

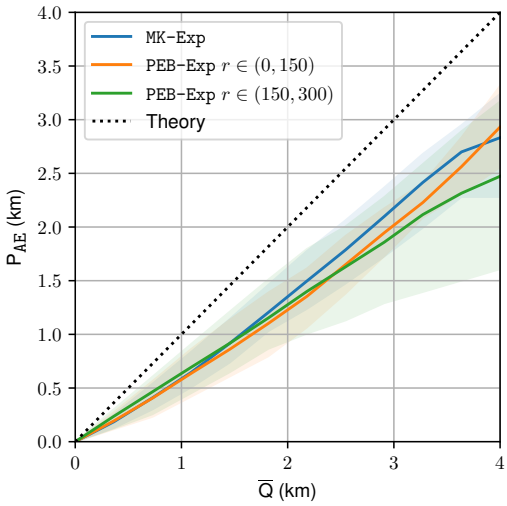
In *non-sporadic* location privacy, our experiments show that PEB-LPPMs can outperform optimal Markov LPPMs when the user's real locations are not highly correlated (i.e., Brightkite and Gowalla datasets). When there are high dependencies between the real locations (i.e., TaxiCab data with location reports every 5 minutes), PEB-LPPMs perform worse than optimal Markov designs because they cannot capture these correlations. This could be addressed in future work by developing PEB-LPPMs based on *blank-slate Markov* models. These PEB-



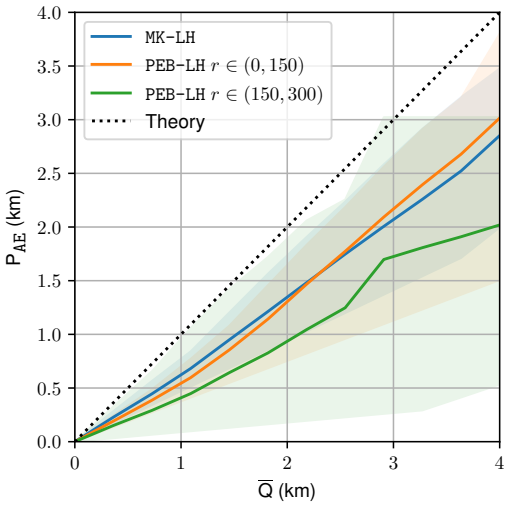
(a) Brightkite, `Exp`.



(b) Brightkite, `LH`.

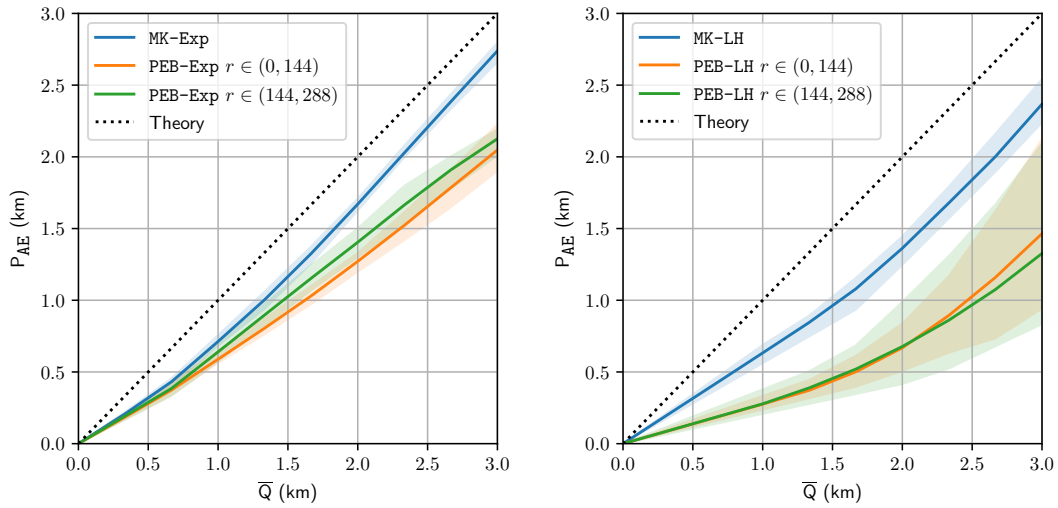


(c) Gowalla, `Exp`.



(d) Gowalla, `LH`.

Figure 5.10: Experiment MK: Performance of PEB-LPPMs versus `MK-LPPM` in Brightkite and Gowalla datasets. `MK-LPPM` have been trained with the rich data.



(a) TaxiCab, `Exp`.

(b) TaxiCab, `LH`.

Figure 5.11: Experiment MK: Performance of PEB-LPPMs versus `MK-LPPM` in TaxiCab dataset, with one day of training. `MK-LPPM` have been trained with the scarce data (a single-day trace).

LPPMs would re-estimate the Markov transition matrix on-the-fly using released locations and taking road restrictions into account.

On another note, in this chapter we only use the average adversary error to measure privacy, and do not show the performance improvements of other privacy metrics that we have seen in the previous chapter, i.e., the conditional entropy and geo-indistinguishability. We do this for simplicity. Throughout our evaluation in this chapter (Figs. 5.9-5.11) we have seen that, in many scenarios, PEB-LPPMs outperform hardwired-based LPPMs. The underlying reason of this improvement is that the mobility profile that PEB-LPPMs estimate a-posteriori characterizes the actual user mobility better than the hardwired models. Thus, we can expect that PEB-LPPMs will also outperform hardwired LPPMs in these scenarios in terms of other privacy metrics, since they are more tailored to the actual user behavior in the testing data (e.g., `PEB-Exp` will provide more geo-indistinguishability or conditional entropy than `SP-Exp` for the same quality loss, in a sporadic location release scenario).

5.8. Related Work

Early surveys of location privacy attacks, defenses and privacy metrics, by Decker [39] and Krumm [40], do not include any discussion about modeling user mobility.

A first explicit modeling appears in [51], where Shokri et al. propose a framework to evaluate location privacy mechanisms. In their framework instantiation, they consider a Markov hardwired model for user mobility, and in their evaluation they effectively merge training and testing sets. A number of follow-ups also hardwire the mobility model using the evaluation data itself. In sporadic location privacy, this methodology was used to design and evaluate LPPMs according to different privacy notions. First, it was used to find optimal LPPMs in terms of the average adversary error, either by reporting individual locations [46], using dummy check-ins [90], or in combination with geo-ind guarantees [72]. Second, hardwired user mobility models are used to obtain utility improvements and derive optimal geo-indistinguishability LPPMs [65], or to evaluate a semantic variation of this notion [66]. Non-sporadic location privacy works also hardwired their mobility models on the evaluation data, and typically adopt a Markov model for user mobility to account for temporal correlations [64, 91].

Chatzikokolakis et al. are the first to explicitly separate data used to design LPPMs and to evaluate them [67], in the context of geo-indistinguishability. However, they do not quantify the privacy gap between theoretical design and empirical evaluation in a testing set.

To the best of our knowledge, our work is the first to evaluate previous optimal LPPMs by considering a separation between training and testing data. We find out that LPPMs perform worse than previously reported results [46, 91–93] when empirically evaluated in a testing set. We also propose a blank-slate model for user mobility, which allows us to design LPPMs that *learn* the model parameters during the evaluation. We are not aware of other blank-slate models in the literature, although the mobility profile estimation carried by PEB-LPPMs is similar to the problem of estimating a distribution from noisy data in privacy-preserving data mining [98].

5.9. Conclusions

Previous strategies to design Location Privacy-Preserving Mechanisms (LPPMs) assume that training data can completely characterize user mobility behavior, and hardwire this information in the mechanism itself. We demonstrate how this design decision overestimates the privacy offered by these designs when the users' mobility profile deviates from the training set characteristics.

We propose to use blank-slate models for user mobility that treat the mobility profile as an unknown variable that has to be *learned*. We leverage a sporadic blank-slate model to propose a new family of defense techniques, PEB-LPPMs, that adapt to the user behavior using past obfuscated queries. We compare our proposal to hardwired LPPMs, and show that PEB-LPPMs improve the privacy

except in continuous location release scenarios where user locations are highly correlated.

The problem identified in this chapter is not unique to the location privacy domain. More generally, to build privacy enhancing technologies that provide strong privacy guarantees in real cases, we have to embrace that training information cannot always fully capture real user behavior. We believe that blank-slate models, that incorporate the uncertainty about real user behavior, are a promising approach to improve the protection provided by privacy mechanisms not only in location privacy but in a broader type of privacy problems.

Appendix

5.A. Performance of Memoryless LPPMs in the Hardwired Model.

Consider the full-type LPPM $f(z^r|\mathbf{z}^{r-1}, \mathbf{x}^r)$, and a memoryless-type LPPM that we denote by f^*, defined as

$$f^*(z^r|x^r) \doteq \sum_{\substack{\mathbf{x}^{r-1} \\ \in \mathcal{X}^{r-1}}} \sum_{\substack{\mathbf{z}^{r-1} \\ \in \mathcal{Z}^{r-1}}} p(\mathbf{x}^{r-1}, \mathbf{z}^{r-1}|x^r) \cdot f(z^r|\mathbf{z}^{r-1}, \mathbf{x}^r)\,. \tag{5.22}$$

The average loss of f and f^* is the same, i.e., $\overline{\mathrm{Q}}(f, r) = \overline{\mathrm{Q}}(f^*, r)$ due to the linearity of this metric. Then, by proving that f^* does not achieve less privacy than f, we prove that the privacy and quality loss trade-off of f^* is not worse than that of f. For these proofs, we use p^* to denote the probabilities referred to the case where the LPPM used is f^*. Also, we use $\mathbf{z}^{-s} \doteq [z^1, z^2, \cdots, z^{s-1}, z^{s+1}, \cdots, z^r]$.

Our goal is to prove that $\min_h \mathbf{P}_{\mathtt{AE}}(f, h, r, s) \leq \min_h \mathbf{P}_{\mathtt{AE}}(f^*, h, r, s)$, i.e., that f^* does not achieve less privacy than f against an optimal adversary that minimizes $\mathbf{P}_{\mathtt{AE}}$:

$$\begin{aligned}
\min_h \mathbf{P}_{\mathtt{AE}}(f, h, r, s) &= \sum_{\mathbf{z}^r \in \mathcal{Z}^r} \min_{\hat{x}^s} \left[\sum_{x^s \in \mathcal{X}} \pi(x^s) p(\mathbf{z}^r|x^s) d_P(x^s, \hat{x}^s) \right] \\
&\overset{(a)}{\leq} \sum_{z^s \in \mathcal{Z}} \min_{\hat{x}^s} \left[\sum_{\mathbf{z}^{-s} \in \mathcal{Z}^{-s}} \sum_{x^s \in \mathcal{X}} \pi(x^s) p(\mathbf{z}^r|x^s) d_P(x^s, \hat{x}^s) \right] \\
&= \sum_{z^s \in \mathcal{Z}} \min_{\hat{x}^s} \left[\sum_{x^s \in \mathcal{X}} \pi(x^s) f^*(z^s|x^s) d_P(x^s, \hat{x}^s) \right] \\
&\overset{(b)}{=} \sum_{\mathbf{z}^r \in \mathcal{Z}^r} \min_{\hat{x}^s} \left[\sum_{x^s \in \mathcal{X}} \pi(x^s) p(\mathbf{z}^{-s}) f^*(z^s|x^s) d_P(x^s, \hat{x}^s) \right] \\
&= \min_h \mathbf{P}_{\mathtt{AE}}(f^*, r, s)\,.
\end{aligned}$$

Step (a) comes from splitting the summation over $\mathbf{z}^r$ into two summations: one over z^s and the other over the complement. Then, computing the summation (over $\mathbf{z}^{-s}$) of the minima over $\hat{x}^s$ is smaller or equal than computing the minimum of the summation. Step (b) follows from the fact that $\mathbf{z}^{-s}$ is independent of z^s and x^s in the hardwired model and with a memoryless LPPM f^*.

5.B. Convergence of the EM Sequence to the MLE of the Mobility Profile.

We prove the convergence of the EM iteration in (5.20) to the maximum likelihood estimator of the mobility profile, *for memoryless and output-based LPPMs* only. Let $\mathcal{P}$ be the probability simplex, i.e., the set of valid mobility profiles $\mathcal{P} \doteq \{\pi | \sum_{i=1}^{|\mathcal{X}|} \pi_i = 0, \quad \pi_i \geq 0\}$. Then, the MLE is

$$\hat{\pi}^r_{ML} = \operatorname*{argmax}_{\pi \in \mathcal{P}} \log p(\mathbf{z}^r|\pi)\,. \tag{5.23}$$

In [98,100], authors show that if the likelihood function (i.e., $\log p(\mathbf{z}^r|\pi)$) has a unique global maximum over $\mathcal{P}$ and the derivatives $\partial Q(\pi, \pi^t)/\partial\pi$ are continuous over π and π^t, then any EM sequence $\{\pi^0, \pi^1, \pi^2, \cdots\}$ computed as in (5.20) converges to the unique global maximum $\hat{\pi}^r_{ML}$. We now prove that our problem meets these requirements, and refer to [98, 100] for the complete details of the proof.

First, we prove that $\log p(\mathbf{z}^r|\pi)$ is strictly concave and has a unique global maximum over $\mathcal{P}$. By definition, it is easy to see that $\mathcal{P}$ is convex, i.e., given two profiles $\pi, \pi' \in \mathcal{P}$, we can check that $\pi'' \doteq \lambda\pi + (1-\lambda)\pi' \in \mathcal{P}$ for $\lambda \in [0, 1]$. On the other hand, we can write $\log p(\mathbf{z}^r|\pi) = \sum_{s=1}^{r} \log p(z^s|\mathbf{z}^{s-1}, \pi)$ and show that

$$p(z^s|\mathbf{z}^{s-1}, \pi) = \sum_{i=1}^{|\mathcal{X}|} f(z^s|\mathbf{z}^{s-1}, x^s = x_i) \cdot \pi_i\,, \tag{5.24}$$

where $f(z^s|\mathbf{z}^{s-1}, x^s = x_i)$ is given by the LPPM (it does not require π for its computation, since it is an output-based LPPM). This means that $p(z^s|\mathbf{z}^{s-1}, \pi)$ is linear with π, and therefore $\log p(z^s|\mathbf{z}^{s-1}, \pi)$ is strictly concave. This implies that $\log p(\mathbf{z}^r|\pi)$ is also strictly concave, since it is the sum of strictly concave functions. Since $\mathcal{P}$ is a convex set, then $\log p(\mathbf{z}^r|\pi)$ has a unique global maximum over $\mathcal{P}$.

On the other hand, it is easy to see that the derivatives $\partial Q(\pi, \pi^t)/\partial\pi$ are continuous in π and π^t (note that $\pi_i \in [0, 1]$), which concludes the proof.

The proof for memoryless LPPMs is the same, since they are a sub-type of output-based LPPMs.

Chapter 6

Conclusions and Future Work

In this thesis, we used signal processing tools to develop privacy enhancements for electronic services. We studied two privacy-preserving technologies: mix-based anonymous communication systems, that protect against meta-data leakage, and perturbation-based location privacy mechanisms, that protect against an adversarial location-based service provider.

Our first contribution in mix-based anonymous communication systems is a methodology to allocate dummy traffic so as to maximize the privacy that the mix provides to its users (Chapter 2). We derived a closed-form expression of the anonymity of the users in terms of the system parameters, and used it to study how to optimally allocate dummies so as to achieve specific privacy goals. We illustrated the usefulness of this methodology by deriving two optimal dummy allocation strategies: one that increases the protection of all the users in the system by a constant factor, and another one that maximizes the minimum protection of all the pairwise relations between users in the system.

Then, we studied optimal message delay strategies in pool mixes under realistic user behavior (Chapter 3). We first proposed a behavioral model that accounts for realistic traits, such as the fact that some users spread the messages they send among multiple recipients, while others keep long conversations with a particular recipient before switching to another one. We validated this model with real data, and used it to obtain an expression for the anonymity of the users against the state-of-the-art profiling attack. Using this expression, we found the distribution of the delay of the messages inside the pool mix that maximizes user privacy, and proposed other sub-optimal but easier-to-compute distributions. Our experiments with real data confirm that our proposed delay strategies outperform the state-of-the-art design (i.e., the binomial pool mix).

In the field of location privacy, our work identifies flaws in the approach that is typically followed to design and evaluate Location Privacy Preserving Mechanisms (LPPMs). First, we studied the *metrics* that are used to assess LPPM

performance (Chapter 4). We show that judging LPPMs based on a single privacy and utility metric is misleading, and provide an example of an LPPM that is optimal according to the state-of-the-art location privacy metrics, but is clearly unsuitable for the users' needs. We claim that we must consider privacy and utility as multi-dimensional notions, and advocate for using the conditional entropy as a complementary privacy metric. We develop a (quasi-)optimal LPPM that maximizes the conditional entropy, and evaluate it together with previous proposals in terms of different privacy and utility notions. Our experiments confirm that no mechanism fares well in all the metrics, and that judging an LPPM based on a single metric gives a false perception of privacy.

Finally, in Chapter 5 we studied how to design LPPMs that protect users with unknown mobility behavior. We found that most of the previous works build LPPMs using mobility models *hardwired* on the training data. We showed that these LPPMs are overfitted to the training data, and thus perform below the theoretical expectations of LPPM designers when evaluated on a different testing data. In order to build LPPMs for users whose behavior is not represented by the training data, we proposed a *blank-slate* mobility model. This model is not determined by the training data, but adapts a-posteriori to the actual user behavior as she queries the location-based service provider. We evaluated LPPMs developed with blank-slate models, called PEB-LPPMs, and showed that they improve over hardwired LPPMs when the training data does not capture well the actual users' behavior.

This thesis demonstrates the advantages of following a statistical approach towards designing privacy-preserving systems. Contrary to heuristic or machine learning approaches, our contributions [19, 34, 35, 37, 38, 76, 81, 92, 97] are backed up not only by empirical results, but also by theoretical foundations that ensure that our privacy improvements are effective as long as the user behavioral models that we considered hold in practice.

6.1. Future Research Lines

Even though the two parts of the thesis deal with privacy problems that are very different, the statistical models that we used in both parts are surprisingly similar. Indeed, we can abstract both models in a single one, where there are N users sending messages to M possible destinations through a privacy-preserving channel (Fig. 6.1). An adversary observes the messages sent and received by both parties and wants to either estimate the sending profile of the user (this is the profiling adversary that we assumed in the first part of the thesis) or learn to which destination an input message was headed (this corresponds to the adversary of the second part of the thesis). The privacy-preserving channel can perform different operations that increase the privacy of the user, at some cost: the channel can

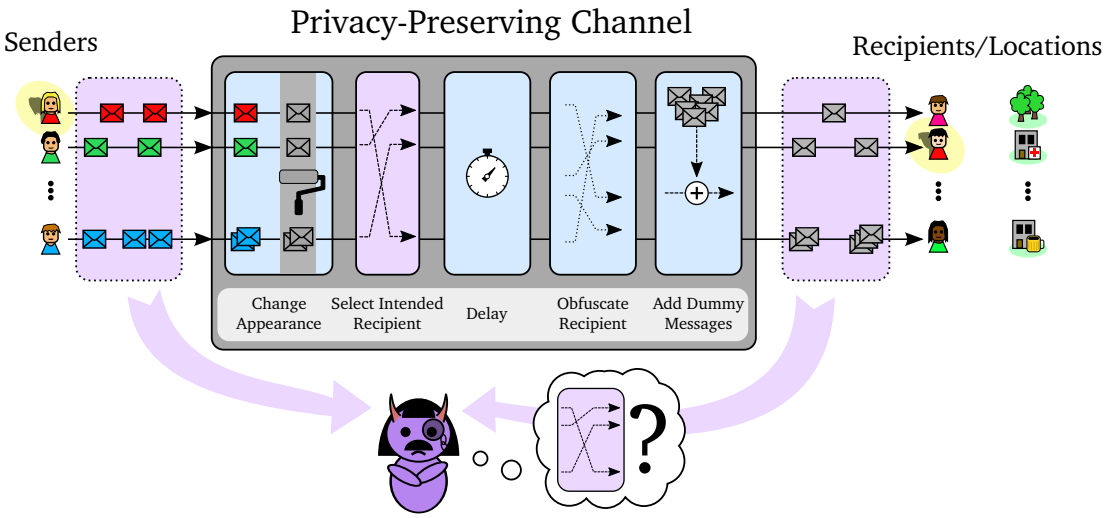


Figure 6.1: General system model of privacy-preserving communication channels that accommodates a wide variety of privacy scenarios, such as the mix-based anonymous communication system and the obfuscation-based location privacy model that we studied in this thesis.

delay messages, at the expense of communication latency; obfuscate or change the recipient of a message, at the expense of service utility; and add dummy messages, at the expense of communication overhead or bandwidth.

Mixes provide anonymity by adding delay (and, possibly dummies), but do not allow recipient obfuscation (changing the recipient of a message contradicts the purpose of the communication, i.e., results in a total loss of service utility). User-centric location privacy mechanisms, however, do not benefit from delay (since they are implemented on the user's device, and thus cannot blend messages from several users together to confuse the adversary), but allow obfuscation of the destination (i.e., the location) and also benefit from dummy messages (as shown in other works [53, 54, 90]).

The general model in Fig. 6.1 accommodates these two privacy scenarios and we believe it could fit many other problems. This opens many research opportunities, as the statistical techniques that we used in this thesis can be used to develop solutions to other privacy problems that fit this general model.

6.1.1. Other Future Lines.

The contents of the chapters in this thesis are based on our research in chronological order. We believe that we could improve the contributions of our earlier works using the lessons that we learned afterwards.

First, in our study of optimal dummy allocation strategies (Chapter 2), we assumed a simple user behavioral model (the traffic follows a Poisson distribution) that does not hold in real scenarios. We refined this model to account for more realistic user behaviors in our study of delay characteristics (Chapter 3). We would like to take a look back at dummy allocation strategies under realistic user behavior. This is specially challenging, since generating dummy traffic mimicking real behavior is complicated (due to inherent correlations between real messages, e.g., users reply to messages, only send messages when they are available, etc.).

In our work in mixes, we measure privacy as the mean squared estimation error of the users' sending profiles. Then, in our work on location privacy (Chapter 4), we challenged the practice of using a single privacy metric to assess the performance of privacy-preserving techniques. We would like to study how this multi-dimensional privacy notion applies to mix-based anonymous communication systems.

Finally, in Chapters 3 to 4, we trained the users' behavioral models on the same data that we used for evaluation (this is common practice in the related works as well). However, as we discussed in Chapter 5, this leads to designs of privacy-preserving mechanisms that are overfitted to the training data and perform much worse in practice. We would like to evaluate our improvements to mix-based anonymous communication systems using different training and testing data.

Other future research lines that we envision are:

- The methodology of our work in mixes relies on the performance analysis of LSDA, since it is the best-performing profiling attack that can be applied to pool mixes. However, we know that LSDA is sub-optimal, so it would be interesting to find the optimal profiling attack in mixes and test if our findings still hold.

- We used *estimation theory* tools in our analysis of mix-based anonymous communication systems, and derived designs that protect against an adversary that tries to *estimate* the users' sending profiles. However, we could also leverage *decision theory* techniques to study the protection of the users against an adversary that wants to make a *decision* (e.g., an adversary that wants to decide whether or not a particular sender is exchanging messages with any receiver within a set).

- In our work in location privacy, we found that LPPM performance varies considerably between the users within the same dataset. A possible line of future work would be to identify which user mobility traits affect this variation in LPPM performance, in terms of both the average adversary error privacy metric and the conditional entropy metric. This way, we could

determine traits that make users specially vulnerable against the service provider, and modify their LPPMs accordingly.

- Our sporadic blank-slate models perform better than hardwired LPPMs when the user's locations are not highly correlated. However, when they are (e.g., Markov models), sporadic blank-slate models cannot compete in many cases against hardwired Markov-based LPPMs. The reason for this is that hardwired Markov models account for the map mobility restrictions (roads, turns, traffic lights) while the sporadic blank-slate models do not. The next step in this research line would be to develop *Markov blank-slate models* that account for these location correlations and also learn the user behavior on-the-fly.

Bibliography

[1] Matthew Edman and Bülent Yener. On anonymity in an electronic society: A survey of anonymous communication systems. *ACM Computing Surveys*, 42(1), 2010.

[2] Fatemeh Shirazi, Milivoj Simeonovski, Muhammad Rizwan Asghar, Michael Backes, and Claudia Diaz. A survey on routing in anonymous communication protocols. *ACM Computing Surveys (CSUR)*, 51(3):51, 2018.

[3] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[4] Ulf Möller, Lance Cottrell, Peter Palfrader, and Len Sassaman. Mixmaster Protocol — Version 2. IETF Internet Draft, July 2003.

[5] George Danezis, Roger Dingledine, and Nick Mathewson. Mixminion: Design of a type iii anonymous remailer protocol. In *IEEE Symposium on Security and Privacy*, pages 2–15. IEEE Computer Society, 2003.

[6] George Danezis. Mix-networks with restricted routes. In *Privacy Enhancing Technologies*, pages 1–17. 2003.

[7] Michael K Reiter and Aviel D Rubin. Crowds: Anonymity for web transactions. *ACM transactions on information and system security (TISSEC)*, 1(1):66–92, 1998.

[8] Arjun Nambiar and Matthew Wright. Salsa: a structured approach to large-scale anonymity. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 17–26. ACM, 2006.

[9] Paul Syverson, Roger Dingledine, and Nick Mathewson. Tor: The second-generation onion router. In *Usenix Security*, 2004.

[10] Ania M Piotrowska, Jamie Hayes, Tariq Elahi, Sebastian Meiser, and George Danezis. The loopix anonymity system. In *26th USENIX Security Symposium, USENIX Security*, pages 16–18, 2017.

[11] Alan Mislove, Gaurav Oberoi, Ansley Post, Charles Reis, Peter Druschel, and Dan S Wallach. Ap3: Cooperative, decentralized anonymous communication. In *Proceedings of the 11th workshop on ACM SIGOPS European workshop*, page 30. ACM, 2004.

[12] Dogan Kesdogan, Jan Egner, and Roland Büschkes. Stop-and-Go-MIXes providing probabilistic anonymity in an open system. In *Information Hiding*, pages 83–98. 1998.

[13] George Danezis. The traffic analysis of continuous-time mixes. In *Privacy Enhancing Technologies*, pages 35–50. 2005.

[14] Claudia Diaz and Andrei Serjantov. Generalising mixes. In *Privacy Enhancing Technologies*, pages 18–31. 2003.

[15] Oliver Berthold, Andreas Pfitzmann, and Ronny Standtke. The disadvantages of free mix routes and how to overcome them. In *Designing Privacy Enhancing Technologies*, pages 30–45. Springer, 2001.

[16] Roger Dingledine, Michael J Freedman, David Hopwood, and David Molnar. A reputation system to increase mix-net reliability. In *International Workshop on Information Hiding*, pages 126–141. Springer, 2001.

[17] Andreas Pfitzmann, Birgit Pfitzmann, and Michael Waidner. Isdn-mixes: Untraceable communication with very small bandwidth overhead. In *Kommunikation in verteilten Systemen*, pages 451–463. Springer, 1991.

[18] Claudia Diaz and Bart Preneel. Taxonomy of mixes and dummy traffic. In *Working Conference on Privacy and Anonymity in Networked and Distributed Systems*, pages 215–230. Kluwer Academic Publishers, 2004.

[19] Simon Oya, Fernando Pérez-González, and Carmela Troncoso. Design of pool mixes against profiling attacks in real conditions. *IEEE/ACM Transactions on Networking*, 24(6):3662–3675, 2016.

[20] Oliver Berthold and Heinrich Langos. Dummy traffic against long term intersection attacks. In Roger Dingledine and Paul F. Syverson, editors, *Privacy Enhancing Technologies Workshop*, volume 2482 of *LNCS*, pages 110–128. Springer, 2002.

[21] Nayantara Mallesh and Matthew Wright. Countering statistical disclosure with receiver-bound cover traffic. In *12th European Symposium on Research in Computer Security*, pages 547–562, 2007.

[22] Nick Mathewson and Roger Dingledine. Practical traffic analysis: Extending and resisting statistical disclosure. In *4th Workshop on Privacy Enhancing Technologies*, pages 17–34, 2004.

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

[24] Dogan Kesdogan, Dakshi Agrawal, and Stefan Penz. Limits of anonymity in open environments. In *5th Workshop on Information Hiding*, pages 53–69, 2002.

[25] Dogan Kesdogan and Lexi Pimenidis. The hitting set attack on anonymity protocols. In *6th Workshop on Information Hiding*, pages 326–339, 2004.

[26] Dang Vinh Pham, Joss Wright, and Dogan Kesdogan. A practical complexity-theoretic analysis of mix systems. In V. Atluri and C. Diaz, editors, *16th European Symposium on Research in Computer Security*, volume 6879 of *LNCS*, pages 508–527. Springer, 2011.

[27] George Danezis and Andrei Serjantov. Statistical disclosure or intersection attacks on anonymity systems. In *6th Workshop on Information Hiding*, pages 293–308, 2004.

[28] George Danezis, Claudia Diaz, and Carmela Troncoso. Two-sided statistical disclosure attack. In *7th Symposium on Privacy Enhancing Technologies*, pages 30–44, 2007.

[29] Nayantara Mallesh and Matthew Wright. The reverse statistical disclosure attack. In R. Böhme, P. W. L. Fong, and R. Safavi-Naini, editors, *12th Information Hiding Conference*, volume 6387 of *LNCS*, pages 221–234. Springer, 2010.

[30] Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In *2nd Workshop on Privacy Enhancing Technologies*, pages 41–53, 2002.

[31] Carmela Troncoso, Benedikt Gierlichs, Bart Preneel, and Ingrid Verbauwhede. Perfect matching disclosure attacks. In *8th Symposium on Privacy Enhancing Technologies*, pages 2–23, 2008.

[32] George Danezis and Carmela Troncoso. Vida: How to use Bayesian inference to de-anonymize persistent communications. In *9th Privacy Enhancing Technologies Symposium*, pages 56–72, 2009.

[33] Fernando Pérez-González and Carmela Troncoso. Understanding statistical disclosure: A least squares approach. In Matthew Wright and Simone Fischer-Hübner, editors, *Privacy Enhancing Technologies Symposium*, volume 7384 of *LNCS*, pages 38–57. Springer-Verlag, 2012.

[34] Simon Oya, Carmela Troncoso, and Fernando Pérez-González. Meet the family of statistical disclosure attacks. *IEEE Global Conference on Signal and Information Processing*, page 4p, 2013.

[35] Fernando Perez-Gonzalez, Carmela Troncoso, and Simon Oya. A least squares approach to the static traffic analysis of high-latency anonymous communication systems. *IEEE Transactions on Information Forensics and Security*, 9(9):1341–1355, 2014.

[36] Fernando Pérez-González and Carmela Troncoso. A least squares approach to user profiling in pool mix-based anonymous communication systems. In *IEEE Workshop on Information Forensics and Security*, pages 115–120, 2012.

[37] Simon Oya, Carmela Troncoso, and Fernando Pérez-González. Understanding the effects of real-world behavior in statistical disclosure attacks. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 72–77, 2014.

[38] Simon Oya, Fernando Pérez-González, and Carmela Troncoso. Filter design for delay-based anonymous communications. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 2107–2111. IEEE, 2017.

[39] Michael Decker. Location privacy-an overview. In *Proc. of the International Conference on Mobile Business (ICMB)*, pages 221–230. IEEE, 2008.

[40] John Krumm. A survey of computational location privacy. *Personal and Ubiquitous Computing*, 13(6):391–399, 2009.

[41] Matt Duckham and Lars Kulik. Location privacy and location-aware computing. In *Dynamic and Mobile GIS*, pages 63–80. CRC Press, 2006.

[42] Ge Zhong, Ian Goldberg, and Urs Hengartner. Louis, lester and pierre: Three protocols for location privacy. In *International Workshop on Privacy Enhancing Technologies*, pages 62–76. Springer, 2007.

[43] Anh Pham, Kévin Huguenin, Igor Bilogrevic, Italo Dacosta, and Jean-Pierre Hubaux. Securerun: Cheat-proof and private summaries for location-based activities. *IEEE Transactions on Mobile Computing*, 15(8):2109–2123, 2016.

[44] Anh Pham, Italo Dacosta, Guillaume Endignoux, Juan Ramon Troncoso Pastoriza, Kévin Huguenin, and Jean-Pierre Hubaux. Oride: A privacy-preserving yet accountable ride-hailing service. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*, pages 1235–1252, 2017.

[45] Miguel E Andrés, Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Proc. of Computer and Communications Security (CCS)*, pages 901–914. ACM, 2013.

[46] Reza Shokri, George Theodorakopoulos, Carmela Troncoso, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. Protecting location privacy: optimal strategy against localization attacks. In *Proc. of Computer and Communications Security (CCS)*, pages 617–627. ACM, 2012.

[47] Gabriel Ghinita, Panos Kalnis, and Spiros Skiadopoulos. Prive: anonymous location-based queries in distributed mobile systems. In *Proceedings of the 16th international conference on World Wide Web*, pages 371–380. ACM, 2007.

[48] Mohamed F Mokbel, Chi-Yin Chow, and Walid G Aref. The new casper: Query processing for location services without compromising privacy. In *Proceedings of the 32nd international conference on Very large data bases*, pages 763–774. VLDB Endowment, 2006.

[49] Bugra Gedik and Ling Liu. Location privacy in mobile systems: A personalized anonymization model. In *Distributed Computing Systems, 2005. ICDCS 2005. Proceedings. 25th IEEE International Conference on*, pages 620–629. IEEE, 2005.

[50] Bhuvan Bamba, Ling Liu, Peter Pesti, and Ting Wang. Supporting anonymous location queries in mobile environments with privacygrid. In *Proceedings of the 17th international conference on World Wide Web*, pages 237–246. ACM, 2008.

[51] Reza Shokri, George Theodorakopoulos, Jean-Yves Le Boudec, and Jean-Pierre Hubaux. Quantifying location privacy. In *Proc. of Security and Privacy (S&P)*, pages 247–262. IEEE, 2011.

[52] Hidetoshi Kido, Yutaka Yanagisawa, and Tetsuji Satoh. An anonymous communication technique using dummies for location-based services. In *ICPS'05. Proceedings. International Conference on Pervasive Services, 2005.*, pages 88–97. IEEE, 2005.

[53] Hua Lu, Christian S. Jensen, and Man Lung Yiu. Pad: privacy-area aware, dummy-based location privacy in mobile services. In *ACM International Workshop on Data Engineering for Wireless and Mobile Access*, pages 16–23. ACM, 2008.

[54] Tun-Hao You, Wen-Chih Peng, and Wang-Chien Lee. Protecting moving trajectories with dummies. In *International Conference on Mobile Data Management*, pages 278–282, 2007.

[55] Pierangela Samarati. Protecting respondents identities in microdata release. *IEEE transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[56] Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proc. of the International Conference on Mobile Systems, Applications and Services (MobiSys)*, pages 31–42. ACM, 2003.

[57] Claudio Bettini, X Sean Wang, and Sushil Jajodia. Protecting privacy against location-based personal identification. In *Workshop on Secure Data Management*, pages 185–199. Springer, 2005.

[58] 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.

[59] Ge Zhong and Urs Hengartner. A distributed k-anonymity protocol for location privacy. In *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, pages 1–10. IEEE, 2009.

[60] Kar Way Tan, Yimin Lin, and Kyriakos Mouratidis. Spatial cloaking revisited: Distinguishing information leakage from anonymity. In *International Symposium on Spatial and Temporal Databases*, pages 117–134. Springer, 2009.

[61] Toby Xu and Ying Cai. Feeling-based location privacy protection for location-based services. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 348–357. ACM, 2009.

[62] Reza Shokri, Carmela Troncoso, Claudia Diaz, Julien Freudiger, and Jean-Pierre Hubaux. Unraveling an old cloak: k-anonymity for location privacy. In *Proceedings of the 9th annual ACM workshop on Privacy in the electronic society*, pages 115–118. ACM, 2010.

[63] Reza Shokri, Julien Freudiger, Murtuza Jadliwala, and Jean-Pierre Hubaux. A distortion-based metric for location privacy. In *Proceedings of the 8th ACM workshop on Privacy in the electronic society*, pages 21–30. ACM, 2009.

[64] Berker Ağır, Kévin Huguenin, Urs Hengartner, and Jean-Pierre Hubaux. On the privacy implications of location semantics. *Proc. of Privacy Enhancing Technologies (PETS)*, 2016(4):165–183, 2016.

[65] Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Optimal geo-indistinguishable mechanisms for location privacy. In *Proc. of Computer and Communications Security (CCS)*, pages 251–262. ACM, 2014.

[66] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Marco Stronati. Constructing elastic distinguishability metrics for location privacy. *Proc. of Privacy Enhancing Technologies (PETS)*, 2015(2):156–170, 2015.

[67] Konstantinos Chatzikokolakis, Ehab Elsalamouny, and Catuscia Palamidessi. Efficient utility improvement for location privacy. *Proc. of Privacy Enhancing Technologies (PETS)*, 2017(4):308–328, 2017.

[68] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006.

[69] Cynthia Dwork. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*, pages 1–19. Springer, 2008.

[70] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Marco Stronati. A predictive differentially-private mechanism for mobility traces. In *Proc. of Privacy Enhancing Technologies (PETS)*, pages 21–41. Springer, 2014.

[71] Yonghui Xiao and Li Xiong. Protecting locations with differential privacy under temporal correlations. In *ACM Conference on Computer & Communications Security*, pages 1298–1309. ACM, 2015.

[72] Reza Shokri. Privacy games: Optimal user-centric data obfuscation. *Proc. of Privacy Enhancing Technologies (PETS)*, 2015(2):299–315, 2015.

[73] Lei Yu, Ling Liu, and Calton Pu. Dynamic differential location privacy with personalized error bounds. 2017.

[74] Kassem Fawaz and Kang G. Shin. Location privacy protection for smartphone users. In Gail-Joon Ahn, Moti Yung, and Ninghui Li, editors, *ACM SIGSAC Conference on Computer and Communications Security*, pages 239–250. ACM, 2014.

[75] Location guard, 2016. Accessed: 2017-06-12.

[76] Simon Oya, Carmela Troncoso, and Fernando Pérez-González. Is geo-indistinguishability what you are looking for? In *Workshop on Privacy in the Electronic Society*, pages 137–140, 2017.

[77] Claudia Diaz and Bart Preneel. Reasoning about the anonymity provided by pool mixes that generate dummy traffic. In Jessica J. Fridrich, editor, *Workshop on Information Hiding*, volume 3200 of *LNCS*, pages 309–325. Springer-Verlag, 2004.

[78] David Rebollo-Monedero, Javier Parra-Arnau, Claudia Diaz, and Jordi Forné. On the measurement of privacy as an attacker's estimation error. *International Journal of Information Security*, 12(2):129–149, 2013.

[79] George Danezis. Statistical disclosure attacks: Traffic confirmation in open environments. In *Proceedings of Security and Privacy in the Age of Uncertainty*, pages 421–426, Athens, 2003.

[80] Anonymized for submission. A least squares approach to the traffic analysis of high-latency anonymous communication systems. https://www.dropbox.com/s/96pa2c4waxw1ca4/techreport.pdf.

[81] Simon Oya, Carmela Troncoso, and Fernando Pérez-González. Do dummies pay off? limits of dummy traffic protection in anonymous communications. In Emiliano De Cristofaro and StevenJ. Murdoch, editors, *Privacy Enhancing Technologies*, volume 8555 of *Lecture Notes in Computer Science*, pages 204–223. Springer International Publishing, 2014.

[82] Simon Haykin. *Adaptive Filter Theory, 4/e.* Prentice Hall, 2002.

[83] David Rebollo-Monedero, Javier Parra-Arnau, Jordi Forné, and Claudia Diaz. Optimizing the design parameters of threshold pool mixes for anonymity and delay. *Computer Networks*, 67(0):180–200, July 2014.

[84] Simon Oya, Fernando Pérez-González, and Carmela Troncoso. Technical report for id tnet-2015-00294 "optimal delay characteristic when the number of users is comparable to the number of rounds". http://gpsc.uvigo.es/sites/default/files/publications/TechRepToN2016.pdf.

[85] Reza Shokri, Julien Freudiger, Murtuza Jadliwala, and Jean-Pierre Hubaux. A distortion-based metric for location privacy. In Ehab Al-Shaer and Stefano Paraboschi, editors, *ACM Workshop on Privacy in the Electronic Society, WPES*, pages 21–30. ACM, 2009.

[86] Kassem Fawaz, Huan Feng, and Kang G. Shin. Anatomization and protection of mobile apps' location privacy threats. In Jaeyeon Jung and Thorsten Holz, editors, *24th USENIX Security Symposium*, pages 753–768. USENIX Association, 2015.

[87] Changsha Ma and Chang Wen Chen. Nearby friend discovery with geo-indistinguishability to stalkers. *Procedia Computer Science*, 34:352–359, 2014.

[88] Igor Bilogrevic, Kévin Huguenin, Stefan Mihaila, Reza Shokri, and Jean-Pierre Hubaux. Predicting users' motivations behind location check-ins and utility implications of privacy protection mechanisms. In *22nd Network and Distributed System Security Symposium (NDSS)*, 2015.

[89] Thomas M Cover and Joy A Thomas. *Elements of information theory.* John Wiley & Sons, 2012.

[90] Michael Herrmann, Carmela Troncoso, Claudia Diaz, and Bart Preneel. Optimal sporadic location privacy preserving systems in presence of bandwidth constraints. In *Proc. of Workshop on Privacy in the Electronic Society (WPES)*, pages 167–178. ACM, 2013.

[91] Reza Shokri, George Theodorakopoulos, and Carmela Troncoso. Privacy games along location traces: A game-theoretic framework for optimizing location privacy. *Transactions on Privacy and Security (TOPS)*, 19(4):11, 2017.

[92] Simon Oya, Carmela Troncoso, and Fernando Pérez-González. Back to the drawing board: Revisiting the design of optimal location privacy-preserving mechanisms. In *Proc. of Computer and Communications Security (CCS)*, pages 1959–1972. ACM, 2017.

[93] George Theodorakopoulos, Reza Shokri, Carmela Troncoso, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. Prolonging the hide-and-seek game: Optimal trajectory privacy for location-based services. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, pages 73–82. ACM, 2014.

[94] John Krumm. Inference attacks on location tracks. In *International Conference on Pervasive Computing*, volume 4480, pages 127–143. Springer, 2007.

[95] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. Show me how you move and I will tell you who you are. *Transactions on Data Privacy*, 4(2):103–126, 2011.

[96] Reza Shokri, George Theodorakopoulos, George Danezis, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. Quantifying location privacy: the case of sporadic location exposure. In *Proc. of Privacy Enhancing Technologies (PETS)*, pages 57–76. Springer, 2011.

[97] Simon Oya, Carmela Troncoso, and Fernando Pérez-González. Rethinking location privacy for unknown mobility behaviors. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2019.

[98] Dakshi Agrawal and Charu C Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proc. of the Symposium on Principles of Database Systems (PODS)*, pages 247–255. ACM, 2001.

[99] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[100] CF Jeff Wu. On the convergence properties of the em algorithm. *The Annals of Statistics*, pages 95–103, 1983.