2017

Privacy In Multi-Agent And Dynamical Systems

Fragkiskos Koufogiannis
University of Pennsylvania, fkouf@seas.upenn.edu

Privacy In Multi-Agent And Dynamical Systems

Abstract
The use of private data is pivotal for numerous services including location--based ones, collaborative recommender systems, and social networks. Despite the utility these services provide, the usage of private data raises privacy concerns to their owners. Noise--injecting techniques, such as differential privacy, address these concerns by adding artificial noise such that an adversary with access to the published response cannot confidently infer the private data. Particularly, in multi--agent and dynamical environments, privacy--preserving techniques need to be expressive enough to capture time--varying privacy needs, multiple data owners, and multiple data users. Current work in differential privacy assumes that a single response gets published and a single predefined privacy guarantee is provided. This work relaxes these assumptions by providing several problem formulations and their approaches. In the setting of a social network, a data owner has different privacy needs against different users. We design a coalition--free privacy--preserving mechanism that allows a data owner to diffuse their private data over a network. We also formulate the problem of multiple data owners that provide their data to multiple data users. Also, for time--varying privacy needs, we prove that, for a class of existing privacy--preserving mechanism, it is possible to effectively relax privacy constraints gradually. Additionally, we provide a privacy--aware mechanism for time--varying private data, where we wish to protect only the current value of it. Finally, in the context of location--based services, we provide a mechanism where the strength of the privacy guarantees varies with the local population density. These contributions increase the applicability of differential privacy and set future directions for more flexible and expressive privacy guarantees.

Degree Type
Dissertation

Degree Name
Doctor of Philosophy (PhD)

Graduate Group
Electrical & Systems Engineering

First Advisor
George J. Pappas

Keywords
Differential privacy, Dynamical systems, Multi-agent systems, Privacy

Subject Categories
Engineering

PRIVACY IN MULTI–AGENT AND DYNAMICAL SYSTEMS

Fragkiskos Koufogiannis

A DISSERTATION

in

Electrical and Systems Engineering

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2017

Supervisor of Dissertation

George J. Pappas, Professor of Electrical and Systems Engineering

Graduate Group Chairperson

Alejandro Ribeiro, Associate Professor of Electrical and Systems Engineering

Dissertation Committee

Victor M. Preciado, Associate Professor of Electrical and Systems Engineering

Aaron Roth, Associate Professor of Electrical and Systems Engineering

Jesse Walker, Research Professor of Electrical Engineering and Computer Science, Oregon State University

PRIVACY IN MULTI–AGENT AND DYNAMICAL SYSTEMS

In memory of my father

ACKNOWLEDGEMENTS

I wish to deeply thank my adviser George J. Pappas. I am grateful for inviting me to work with him, for his continuous trust and support throughout my studies, for his encouragement to pursuit and develop my own research vision, for both his technical to high–level advice, and for his focus on my personal development.

I sincerely thank Victor Preciado, Aaron Roth, and Jesse Walker for serving in my committee.

I would like to thank my fellow group members and my friends at the University of Pennsylvania and in Philadelphia for the memorable times. I wish to all of them best of luck in their adventures.

Finally, I wish to thank my brother, Ioannis, whose colossal support and monumental guidance cannot be fully described by any adjectives, and my parents, Eleni and Dimitrios, who provided me with such a wonderful environment to grow up in.

ABSTRACT

PRIVACY IN MULTI–AGENT AND DYNAMICAL SYSTEMS

Fragkiskos Koufogiannis

George J. Pappas

The use of private data is pivotal for numerous services including location–based ones, collaborative recommender systems, and social networks. Despite the utility these services provide, the usage of private data raises privacy concerns to their owners. Noise–injecting techniques, such as differential privacy, address these concerns by adding artificial noise such that an adversary with access to the published response cannot confidently infer the private data. Particularly, in multi–agent and dynamical environments, privacy–preserving techniques need to be expressive enough to capture time–varying privacy needs, multiple data owners, and multiple data users. Current work in differential privacy assumes that a single response gets published and a single predefined privacy guarantee is provided. This work relaxes these assumptions by providing several problem formulations and their approaches. In the setting of a social network, a data owner has different privacy needs against different users. We design a coalition–free privacy–preserving mechanism that allows a data owner to diffuse their private data over a network. We also formulate the problem of multiple data owners that provide their data to multiple data users. Also, for time–varying privacy needs, we prove that, for a class of existing privacy–preserving mechanism, it is possible to effectively relax privacy constraints gradually. Additionally, we provide a privacy–aware mechanism for time–varying private data, where we wish to protect only the current value of it. Finally, in the context of location–based services, we provide a mechanism where the strength of the privacy guarantees varies with the local population density. These contributions increase the applicability of differential privacy and set future directions for more flexible and expressive privacy guarantees.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

CHAPTER 1: Introduction

In the Internet of Things (IoT) era, a plethora of applications provide information to the end users as a service. Examples include traffic maps that assist users with navigating congested roads and collaborative recommender systems that suggest buyers potentially interesting products. Despite the utility such services provide, these applications rely on gathering, aggregating, and processing personal data from individuals. For example, traffic maps are typically constructed from users that report their current position and speed, and product are suggested based on the buying habits of previous buyers with similar buying history. This usage of personal data has raised privacy concerns. These concerns lead to the need for strong and formal privacy guarantees for the data owners while still enabling services that rely on accessing private data. In fact, privacy is already sought after* and people oppose when their privacy is blatantly violated†.

Providing privacy is a nontrivial task and, to this end, a variety of approaches that formalize the problem has been proposed. Generally, the problem of protecting privacy is formulated as follows. Given a private input, we wish to evaluate a function and publicly release its output such that the published output does not reveal sensitive parts of the private input. The unifying idea of privacy–preserving approaches is injecting artificial noise or encoding/perturbing the private data such that the system is resilient to inference attacks by a curious adversary. Specifically, an adversary that observes the output of the system should not be able to accurately infer the original private data. Some of these privacy–preserving approaches, which will be discussed in more detail later, include differential privacy —which is employed in this work—, k–anonymity, information–theoretic approaches, game–theoretic ones, unobervability notions, and other.

In most of these approaches the strength of the privacy guarantees can be quantified and,

*Senator requests strong privacy guarantees: https://www.wyden.senate.gov/news/press-releases/wyden-pushes-for-stronger-security-in-collection-of-personal-information

†People boycott clothing company for employing RFIDs for tracking: http://www.boycottbenetton.com/

intuitively, a privacy–utility tradeoff emerges: strong privacy guarantees result in less utility of the public output. In differential privacy, this tradeoff is termed *privacy level*. A important limitation that we identify in the literature and explore in this work is the underlying assumption of a single, fixed privacy level. Specifically, prior work assumes that the privacy level is a designer's choice and that the same output is shared with everyone. In practice, individuals' privacy needs may vary over time, while data owner re–evaluate their privacy preferences. Additionally, these privacy needs also may also depend on the trust level of the end user or may depend on the private data itself. This work focuses on introducing problems and providing approaches where *composite* privacy guarantees are needed and, thus, aiming to broaden the applicability of differential privacy to IoT applications.

1.1. Related Work

Privacy has been identified as an important challenge in developing intelligent infrastructure as pointed out in surveys such as Atzori et al. (2010) and white papers such as Dwork and Pappas (2017).

Preserving privacy is not straightforward. This has become apparent when seemingly privacy–preserving approaches led to significant leaks of privacy. In the infamous case of the Netflix prize (Bennett et al. (2007)), Netflix released the database of users' reviews after scrubbing personal attributes from it and, thus, de–anonymizing it. Nonetheless, Narayanan and Shmatikov (2006) managed to reconstruct the users' identities after correlating the released database with information available from IMDb. Furthermore, Acquisti and Gross (2009) showed that although the Social Security Number (SSN) is deemed a personal data, it can often be predicted from publicly available information. Moreover, Sweeney (1997) provides another inference attack on de–anonymized medical records. The lessons learned from this privacy breach were twofold. First, guaranteeing privacy is not trivial and, secondly, unexpected side information can be catastrophic for privacy.

In the following, we summarize some privacy–preserving approaches that have been explo-

red in the literature. Sweeney (2002) introduced k–anonymity, where an individual should be indistinguishable from $k-1$ other individuals. A similar cloaking technique was explored by Hoh et al. (2007) and Hoh et al. (2008) for traffic monitoring. However, the privacy guarantees provided are fragile when side information becomes available as surveyed by Ohm (2009). Nonetheless, the strength of privacy guarantees is quantified by the parameter k which is assumed to be a designer's choice. Conceptually related to our work, Speranzon and Bopardikar (2016) revisited k–anonymity and explored the evolution of the privacy–preserving mechanism for varying parameter k. Interpreting privacy as unobservability, Pequito et al. (2014) and Fanti et al. (2017) provide privacy–preserving techniques in networks of agents.

Another popular privacy–preserving approach is based on information–theoretic measures. Serjantov and Danezis (2002) and Sankar et al. (2013b) propose adding artificial noise such that the mutual information between private input and the public output is bounded. This approach is especially appealing for smart grid applications where the private input is a signal (Rial and Danezis (2011), Sankar et al. (2013a)). Information–theoretic quantities are also used by Han et al. (2016) and Tanaka et al. (2017) to provide privacy. However, these works typically assume a prior over the private data and provide privacy *in expectation*. Therefore, rare but severe privacy leaks may still occur. Wang et al. (2016) show that overcoming these limitations leads to notions similar to differential privacy.

Differential privacy was introduced by Dwork et al. (2006) and was initially tailored to answering queries on private databases. Differential privacy is formally explained in Section 2. Intuitively, a mechanism is differentially private if small variations of the private input do not change the output significantly. Specifically, the dependency of the probability distribution of the published output on the private data should be bounded. Therefore, a curious adversary cannot accurately infer the private data. The popularity of differential privacy is attributed to its properties. For example, it degrades gracefully in the presence of arbitrary side information, its definition is versatile, and it can often be translated to concrete

application–specific guarantees. On a side note, differential privacy is becoming an industry standard as companies including Google (Erlingsson et al. (2014)) and Apple[‡] deploy it in their products.

In the literature, differential privacy has been explored in combination with every imaginable field. On the theoretical side, Dwork et al. (2006) defined differential privacy, provided the Laplace and the Gaussian mechanism which are used as building blocks for more complicated privacy–preserving mechanisms, and early versions of the composition theorems. Tighter versions of the composition theorems were provided by Kairouz et al. (2017) for general cases. Although relevant to this work, our results in Chapter 3 and Chapter 4 provide better privacy for the cases considered than what composition theorems suggest. Furthermore, several variations of differential privacy have been explored in the literature. For example, Chatzikokolakis et al. (2013) provides a slight reformulation that is appealing for metric spaces and is employed in this work. More recently, Dwork and Rothblum (2016) and Mironov (2017) propose other variations of differential privacy. Although this work is not using these definitions, the concepts are still applicable.

Constructing optimal differential private mechanisms is an important problem. The optimality of the Laplace mechanism is established by Ghosh et al. (2012), Geng and Viswanath (2014), and Wang et al. (2014). Hardt and Talwar (2010) provides asymptotic results for linear queries and Kairouz et al. (2014) prove optimality results under the model of local differential privacy. For our work, the optimality of the Laplace mechanism is utilized in formulating the problems in Chapter 3 and Chapter 4. Additionally, we provide a novel short proof of a result similar to the one by Geng and Viswanath (2014) and Wang et al. (2014).

Closer to applications, existing work provides differentially private versions of existing algorithms and functions. For example, Le Ny and Pappas (2014) build a private version of filters for private input signals, Mo and Murray (2017), Huang et al. (2012), and Katewa

[‡]https://techcrunch.com/2016/06/14/differential-privacy/

et al. (2015) provide private consensus algorithms that guarantee the privacy of the initial states or the communication graph, Han et al. (2014), Hale and Egerstedty (2015), and Huang et al. (2015) propose private optimization algorithms for the case that the objectives or the constraints are private. For results of differential privacy on databases, machine learning, and mechanism design, we refer the reader to the book by Dwork and Roth (2013). The majority of our work is not bound to a specific application and, for example, the results in Chapter 4 add flexibility to existing domain–specific results such as that by Erlingsson et al. (2014).

A key quantity of differential privacy is the so–called privacy level which quantifies the trade–off between the strength of the privacy guarantees and the utility of the published response. The majority of literature work assumes a single privacy level that is a designer's choice. Works that deviate from this assumption include joint differential privacy introduced by Hsu et al. (2016) and personalized/heterogeneous privacy introduced by Alaggan et al. (2016) and Ebadi et al. (2015). These works seem more relevant to the problems explored in Chapter 3 and Chapter 7. Also relevant to this work, methods for choosing an appropriate privacy level have been proposed either before accessing the private data (Ghosh and Roth (2015)) or, more generally, after accessing them (Ligett et al. (2017) builds on the results presented here).

Finally, we mention that cryptographic techniques have been proposed as a means of providing privacy. Works such as those presented by Garcia and Jacobs (2010) and Shoukry et al. (2016) leverage partial homomorphic encryption schemes such as the one introduced by Paillier (1999) and handle only encrypted private data. Although these techniques do not inject noise and, thus, do not suffer from utility degradation, they are computationally expensive, are not robust against arbitrary side information, and are not very versatile. Importantly, the aforementioned works as well as hardware variants of it such as Intel's SGX instruction set extension (Costan and Devadas (2016)) can be viewed as instances of secure multiparty computation systems. As such, they guarantee that the *algorithmic* implemen-

tation does not leak anything about the private data *other than* what the public output already does. However, the output may already reveal too much private information. As such, we view crypto–based approaches as orthogonal to privacy–preserving notions such as differential privacy[§].

1.2. Outline and Contributions

The unifying underlying idea of this work is to extend differential privacy and render it more applicable for dynamical and multi–agent environments such as those found in IoT applications. Specifically, instead of providing a single privacy guarantee, that of protecting a private input given a public output, with a single privacy level, we explore cases where the privacy level changes over time, varies across data users, or depends on the private data itself. A brief description of each of these cases follows.

Diffusing Private Data

In Chapter 3, we introduce the problem of sharing private data with multiple users under different privacy levels. For example, in a social network, a data owner has different privacy needs against different users, i.e. friends are more trusted than strangers and, thus, less privacy is needed. The problem does not decompose across users, since users may collude, exchange information, and, thus, harm the privacy guarantees. In order to address this challenge, we derive a privacy–preserving mechanism that enables private data to be diffused over a network. In particular, each user receives a differentially private proxy of the owner's private data where coalitions of users and de–incentivized. We illustrate our mechanism with two examples: one on synthetic data where the users share their GPS coordinates; and one on a Facebook ego–network where a user shares her infection status. A journal version

[§]This distinction between security and privacy was pointed out in the seminal paper of differential privacy by Dwork et al. (2006).

of this work is found in Koufogiannis and Pappas (2017a).

Gradual Releasing Private Data

In Chapter 4, we introduce the problem of releasing private data under differential privacy when the privacy level is subject to change over time. Existing work assumes that privacy level is determined by the system designer as a fixed value before private data is released. For certain applications, however, users may wish to relax the privacy level for subsequent releases of the same data after either a re-evaluation of the privacy concerns or the need for better accuracy. Specifically, given a database containing private data, we assume that a response y_1 that preserves ϵ_1–differential privacy has already been published. Then, the privacy level is relaxed to ϵ_2, with $\epsilon_2 > \epsilon_1$, and we wish to publish a more accurate response y_2 while the joint response (y_1, y_2) preserves ϵ_2–differential privacy. How much accuracy is lost in the scenario of gradually releasing two responses y_1 and y_2 compared to the scenario of releasing a single response that is ϵ_2–differentially private? Our results consider the more general case with multiple privacy level relaxations and show that there exists a composite mechanism that achieves *no loss* in accuracy.

We consider the case in which the private data lies within $\mathbb{R}^n$ with an adjacency relation induced by the ℓ_1-norm, and we initially focus on mechanisms that approximate identity queries. We show that the same accuracy can be achieved in the case of gradual release through a mechanism whose outputs can be described by a *lazy Markov stochastic process*. This stochastic process has a closed form expression and can be efficiently sampled. Moreover, our results extend beyond identity queries to a more general family of privacy-preserving mechanisms. To this end, we demonstrate the applicability of our tool to multiple scenarios including Google's project RAPPOR, trading of private data, and controlled transmission of private data in a social network. Finally, we derive similar results for the approximated

differential privacy. Koufogiannis et al. (2016) is a published version of this work.

Current State Privacy

In Chapter 5, we introduce the problem of protecting the privacy of *time-varying* sensitive data using differential privacy. Contrary to prior work that considers fixed private data, we wish to design a privacy-preserving mechanism that, at each time and given the observations so far, keeps the *current* state of a dynamical system private. Our work protects dynamical systems from being tracked by an adversary by providing differentially private guarantees.

Specifically, we propose a mechanism which adds artificial noise to (i) the input of the system and (ii) the measurements which are then published. In particular, two scenarios are considered: for a scalar dynamical system under ϵ–differential privacy, we derive a mechanism that, at each time, publishes the most accurate approximation of the current state while preserving privacy. Next, for a general linear system under (ϵ, δ)–differential privacy, we propose a Gaussian–based privacy–preserving mechanism with a quadratic cost. A version of this work is Koufogiannis and Pappas (2017b).

Location–dependent Privacy

In Chapter 6, we consider an application of differential privacy to individuals' GPS positions. Motivated by the need for different privacy levels in different areas —sufficient privacy with less noise can be achieved in urban environments— we propose a generalization of the privacy constraints where the privacy level ϵ is a function of the private data. Next, we establish a connection between this extended notion of differential privacy and a partial differential equation called the *"eikonal equation"*. Exploiting this connection, we propose an algorithm for numerically computing privacy–preserving mechanisms by leveraging existing optimized solvers. The result is demonstrated around the Philadelphia greater area. The

work in this chapter was published in Koufogiannis and Pappas (2016a).

Multi–owner, Multi–user Privacy

In Chapter 7, we focus on designing differentially private mechanisms in a multi–owner multi–user scenario. Specifically, we consider multiple data owners holding a piece of private data and multiple users, each interested in a different function of the entire private data. We formulate the problem of multi–owner multi–user privacy and explore some variations of it. The problem formulation is a generalization of that in Chapter 3, however, the result does not extend for this problem. We focus on owners possessing real–valued private data and users seeking linear functions of this data. Within approximate differential privacy, we propose a Gaussian-based mechanism and express the privacy constraints as constraints on the covariance matrix. Then, we design the mechanism by relaxing these constraints to semi-definite problem. We illustrate our approach in the setting of n agents, each acting both as an data owners and data users. Our approach is numerically evaluated both in terms of efficiency (how much noisier the responses are because of the multiple users) and in terms of incentive to collaborate (how much more information users can gain by collaborating). A conference paper version is Koufogiannis and Pappas (2016b).

CHAPTER 2: Differential Privacy

Differential privacy was introduced by Dwork et al. (2006) and dictates that, whenever private data is accessed, a noisy response is returned. The injected noise is designed to ensure the following two things. First, an adversary that observes the noisy response cannot *confidently infer* the original private data. Secondly, the noisy response can still be used as a surrogate for the exact response *without severe performance degradation.* In order to formally define differential privacy, we need the following ingredients:

- Let $\mathcal{U}$ be the space of private data. Private data can be databases, real numbers, vectors, signals, etc. Also, let $\mathcal{Y}$ be the set of possible responses.

- Let $\mathcal{A} \subseteq \mathcal{U}^2$ be a symmetric binary relation on the space of private data $\mathcal{U}$ called *adjacency relation.* The adjacency relation is a designer's choice and contains the pairs of inputs that need to be rendered almost indistinguishable.

- Let Q be a randomized function, called mechanism, that maps private data $u \in \mathcal{U}$ to a response $y \in \mathcal{Y}$. Equivalently, mechanism Q can be viewed as a function that maps inputs u to distributions over (a sufficiently rich σ–algebra of) $\mathcal{Y}$.

- Let $\epsilon > 0$ and $\delta \in [0, 1]$ be two parameters that capture the strength of the privacy guarantees. Smaller values imply stronger privacy guarantees, while the special case of $\delta = 0$ is of particular importance.

2.1. Definition

The following formal definition introduces the conditions under which a mechanism is differential private.

Definition 1 (Differential Privacy)**.** *Let $\mathcal{U}$ be a set of private data, $\mathcal{A} \subseteq \mathcal{U}^2$ be an adjacency relation and $\mathcal{Y}$ be the set of possible responses. For $\epsilon > 0$ and $\delta \in [0, 1]$, the mechanism*

$Q : \mathcal{U} \to \Delta(\mathcal{Y})$ *is* (ϵ, δ)*-differentially private if*[*]

$$\mathbb{P}(Q\,u \in \mathcal{S}) \leq e^{\epsilon}\,\mathbb{P}(Q\,u' \in \mathcal{S}) + \delta, \quad \forall (u, u') \in \mathcal{A},\ \forall \mathcal{S} \subseteq \mathcal{Y}. \tag{2.1}$$

The special case of $\delta = 0$ is referred to as ϵ–differential privacy or *pure differential privacy* whereas the general case is referred to as (ϵ, δ)–differential privacy or *approximate differential privacy.*

Remark 1. *In Equation* (2.1)*, if we set* $\epsilon = \delta = 0$*, then, we require that* $\mathbb{P}(Q\,u \in \mathcal{S}) = \mathbb{P}(Q\,u' \in \mathcal{S})$*, for all* $(u, u') \in \mathcal{A}$ *and, thus, for meaningful adjacency relations, the response cannot depend on the private data at all. On the other hand, for* $\epsilon \to \infty$ *or* $\delta = 1$*, any mechanism satisfies the privacy constraint in Equation* (2.1)*.*

Remark 2. *We assume the existence of a rich–enough* σ*–algebra* $M \subseteq 2^{\mathcal{Y}}$ *on the set of possible responses* $\mathcal{Y}$*. Then,* $\Delta(\mathcal{Y})$ *denotes the set of probability measures over* $(M, \mathcal{Y})$*. Often, we will let* $\mathcal{Y} = \mathbb{R}^n$ *and, thus, we use the Borel set* $\mathcal{B}^n$*.*

Remark 3. *Let* $y \sim Q\,u$ *be a noisy response produced by the* ϵ*–differentially private mechanism* Q*. For brevity, we say that "output* y *preserves* ϵ*–privacy (or* (ϵ, δ)*–privacy) of the input* u*".*

2.2. Adjacency relations

The adjacency relation $\mathcal{A}$ captures the aspects of the private data u that are deemed sensitive. Consider a scheme with n users, where each user i contributes her real–valued private data $u_i \in \mathbb{R}$, and, thus, a private database $u = [u_1, \ldots, u_n] \in \mathbb{R}^n$ is built. For $\alpha > 0$, an adjacency relation that captures the participation of a single individual to the aggregating scheme is defined as:

$$(u, u') \in \mathcal{A}_{\ell_0} \quad \Leftrightarrow \quad \exists\, j \text{ s.t. } u_i = u'_i,\ \forall i \neq j \text{ and } |u_j - u'_j| \leq \alpha. \tag{2.2}$$

[*]We denote the output of the mechanism Q on the private data u with $Q\,u$ instead of $Q(u)$ in order to simplify expressions.

Adjacency relation $\mathcal{A}_{\ell_0}$ is often replaced by $\mathcal{A}_{\ell_1}$, which is induced by the ℓ_1–norm and is defined as:

$$(u, u') \in \mathcal{A}_{\ell_1} \quad \Leftrightarrow \quad \|u - u'\|_1 \leq \alpha, \tag{2.3}$$

where it holds that $\mathcal{A}_{\ell_0} \subseteq \mathcal{A}_{\ell_1}$.

Another adjacency that is commonly used for private GPS signals (as by Andrés et al. (2013)) and private signals (as by Le Ny and Pappas (2014)) is induced by the ℓ_2–norm defined as follows.

$$(u, u') \in \mathcal{A}_{\ell_2} \quad \Leftrightarrow \quad \|u - u'\|_2 \leq \alpha, \tag{2.4}$$

Adjacency relation $\mathcal{A}_{\ell_2}$ is invariant under rotations which makes it appealing for Euclidean spaces.

Finally, we argue that the adjacency relation needs to capture the domain–specific privacy needs. For instance, Koufogiannis et al. (2014) introduces a more complicated adjacency relation in the context of smart grids and provides an efficient mechanism.

2.3. Properties

An important property of differential privacy is its resilience to post–processing which is the analog of the data processing inequality often used in information–theoretic privacy frameworks (Sankar et al. (2013b)). The property establishes that any post–processing on the output of an (ϵ, δ)–differentially private mechanism cannot weaken the privacy guarantees.

Proposition 2 (Resilience to Post–Processing)**.** *Let $Q : \mathcal{U} \to \Delta(\mathcal{Y})$ be an (ϵ, δ)–differentially private mechanism and $g : \mathcal{Y} \to \mathcal{Z}$ be a possibly randomized function. Then, the mechanism $g \circ Q$ is also (ϵ, δ)–differentially private.*

More complicated mechanisms can be defined from simple ones using the composition theorem.

Proposition 3 (Composition Theorem)**.** *Let mechanisms $Q_1, Q_2 : \mathcal{U} \to \Delta(\mathcal{Y})$ respectively satisfy (ϵ_1, δ_1) and (ϵ_2, δ_2)–differential privacy. Then, the composite mechanism $Q : \mathcal{U} \to \Delta(\mathcal{Y}^2)$ defined by $Q = (Q_1, Q_2)$ is $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$–differentially private.*

Proposition 3 provides privacy guarantees whenever the *same* sensitive data is repeatedly used. Moreover, the resulting privacy level $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$ given by Proposition 3 is an upper bound and can severely over–estimate the actual privacy level. Kairouz et al. (2017) stated tighter bounds for the privacy level degradation. These results are general and are satisfactory when the mechanisms Q_1 and Q_2 are independent and, thus, the joint mechanism (Q_1, Q_2) is less private. Results in Chapter 3 and Chapter 4 avoid unnecessary privacy leaks by strongly correlating the two mechanisms Q_1 and Q_2.

A slightly stronger version of ϵ–differential privacy that is often used when the private data belong to a metric space is defined as a Lipschitz constraint as in Chatzikokolakis et al. (2013). In this case, the adjacency relation is replaced by a metric $d(\cdot, \cdot)$ on $\mathcal{U}$. This notion results in more concise proofs in Chapter 3, Chapter 4, and Chapter 5, and is generalized in Chapter 6. The discrepancy between the original ϵ–differential privacy and its Lipschitz reformulation is discussed in the Appendix A, where the Laplace mechanism is optimal only in the latter case.

Definition 4 (Lipschitz Privacy)**.** *Let $(\mathcal{U}, d)$ be a metric space and $\mathcal{Y}$ be the set of possible responses. For $\epsilon > 0$, the mechanism Q is ϵ–Lipschitz private if the log–likelihood probability is ϵ–Lipschitz in u, i.e.*

$$\left|\ln \mathbb{P}(Q\,u \in \mathcal{S}) - \ln \mathbb{P}(Q\,u' \in \mathcal{S})\right| \leq \epsilon\, d(u, u'), \quad \forall u, u' \in \mathcal{U},\ \forall \mathcal{S} \subseteq \mathcal{Y}. \tag{2.5}$$

Any Lipschitz private mechanism is also differentially private. This implies that our privacy results remain valid within the original framework of differential privacy.

Proposition 5. *For any $\alpha > 0$, an ϵ–Lipschitz private mechanism Q is $\alpha\epsilon$–differentially*

private under the adjacency relation $\mathcal{A}$:

$$(u, u') \in \mathcal{A} \quad \Leftrightarrow \quad d(u, u') \leq \alpha. \tag{2.6}$$

The adjacency relations $\mathcal{A}_{\ell_p}$, $p \in \{1, 2\}$, defined in (2.3) and(2.4) can be captured by the ℓ_1–norm under the notion of Lipschitz privacy; the metric d is $d(u, u') = \|u - u'\|_p$.

Similar to differential privacy, Lipschitz privacy is preserved under post-processing (Proposition 2) and composition of mechanisms is possible (Proposition 3). Compared to differential privacy, Lipschitz privacy is more convenient to work with when the data and adjacency relation are defined on a metric space, which allows for the use of calculus tools.

Remark 4. *Under mild assumptions, the Lipschitz condition* (2.5) *is equivalent to a derivative bound. In particular, for $\mathcal{U} = \mathbb{R}^n$ equipped with the metric induced by the norm $\|\cdot\|$, a mechanism Q is ϵ–Lipschitz private if*

$$\|\nabla_u \ln \mathbb{P}(Q\, u = y)\|_* \leq \epsilon, \tag{2.7}$$

where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$. In practice, we check condition (2.7) *to establish the privacy properties of mechanism Q.*

Finally, two important elementary privacy–preserving mechanisms are the Laplace and the Gaussian one described next. Many domain–specific mechanisms are composed from these two mechanisms. The Laplace mechanism leads to ϵ–privacy and is tailored with optimality results. On the other hand, the Gaussian mechanism results in weaker (ϵ, δ)–privacy (with $\delta > 0$), has lighter tales, and is suitable for traditional linear Gaussian systems.

Theorem 6 (Laplace/Gaussian Mechanism)**.** *Consider the mechanism $Q : \mathcal{U} \to \Delta(\mathbb{R}^n)$ that adds noise to the result of query $q : \mathcal{U} \to \mathbb{R}^n$:*

$$Q(u) = q(u) + V. \tag{2.8}$$

Then,

- *Laplace mechanism:* *for* $n = 1$, *if* $V \sim Lap\left(\frac{\|\Delta q\|_1}{\epsilon}\right)$, *the mechanism* Q *is* ϵ*–differentially private;*
- *Gaussian mechanism:* *if* $V \sim \mathcal{N}\left(0, \frac{\|\Delta q\|_2^2}{\kappa^2(\epsilon,\delta)}\right)$, *the mechanism* Q *is* (ϵ, δ)*–differentially private;*

where $\|\Delta q\|_1 = \max_{(u,u')\in\mathcal{A}} \|q(u) - q(u')\|_1$, $\|\Delta q\|_2 = \max_{(u,u')\in\mathcal{A}} \|q(u) - q(u')\|_2$, *Lap is the Laplace distribution,* $\mathcal{N}$ *is the normal distribution,* $\kappa(\epsilon, \delta) = \frac{2\epsilon}{K+\sqrt{K^2+2\epsilon}}$, *and* $K = \mathcal{Q}^{-1}(\delta)$, *where* $\mathcal{Q}$ *is the tail probability of the normal distribution.*

The quantity $\|\Delta q\|_p$ is the *sensitivity* of the query q and has a key role in differential privacy. As demonstrated in the next chapters, the Laplace and the Gaussian mechanism are often used as building blocks, this work focuses mostly on them. The derived results can, then, be generalized to a broader class of private mechanisms.

CHAPTER 3: Diffusing Private Data

3.1. Motivation and Problem Formulation

In the era of social networks, individuals' profiles include an increasing amount of private information. Although sharing part of this data with close friends under no privacy may be acceptable, privacy concerns arise when less trusted agents use their data. Here, we seek a privacy–preserving technique where the strength of privacy guarantees scales with the trust level of each data user.

Traditionally, these privacy concerns are mitigated by restricting access rights discussed in Section 3.1.2. Instead, we employ differential privacy where the privacy level captures the trust level for each data user. Such an approach allows for a different privacy level against each user. Also, every user in the receives *some* information about each data owner and, thus, global statistics are possible.

System model is presented in Subsection 3.1.1, a baseline approach based on an access–rights scheme is discussed in Subsection 3.1.2, and a concrete problem formulation is presented in Subsection 3.1.3.

3.1.1. System Model

Consider a network represented as a graph G with $|\mathcal{V}| = N$ nodes. For simplicity, we assume that the graph is undirected and unweighted, although this assumption can be removed. Each node $i \in \mathcal{V}$ represents a user and $(i, j) \in \mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the friendship relation between users i and j. Each user owns a private data $u_i \in \mathcal{U}$. Typical examples of private data include:

1. *Timestamps:* let $u_i \in \mathbb{R}$ be a real-valued representation of a timestamp such as date of birth, e.g. *Unix time* Wikipedia (2015) is a popular way of mapping timestamps to integers;

2. *Location:* let $u_i \in \mathbb{R}^2$ be the GPS coordinates of the residence of an individual i;

3. *Binary states:* let $u_i \in \{0, 1\}$ indicate user's i status such as infected or healthy, married or single etc.

Further, we want the severity of the privacy concerns to scale with the distance between two nodes. Typical choices for the distance function $d : \mathcal{V} \times \mathcal{V} \to \mathbb{R}_+$ are as follows:

1. *Shortest path distance:* let d_{ij} be the length of the shortest path connecting nodes i and j;

2. *Resistance distance:* let d_{ij} be the resistance between nodes i and j, where the edges of graph G are associated with unit resistors Babić et al. (2002).

Distance functions that are more suitable for social networks have been proposed in the literature; e.g., see Figure 2 of Liben-Nowell and Kleinberg (2007). In this paper, we assume that these distances are given and we focus on preserving the privacy of the users' data. Moreover, distances d_{ij} may not depend only on the structure of the underlying network but also on the attributes of the nodes. For instance, a family relationship between users i and j may lead to a smaller value of d_{ij}. Further, directed edges (e.g. blocked users) can be also be allowed in social network scenarios.

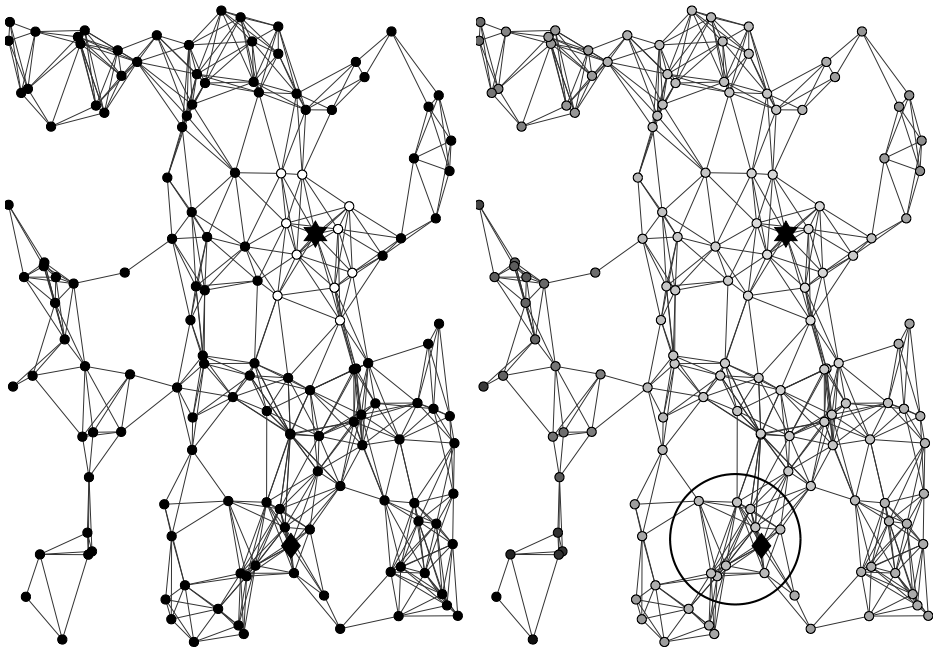
Then, user i generates an approximation y_{ij} of u_i and securely communicates y_{ij} to user j. More specifically, each user i requires her data u_i to be $\epsilon(d_{ij})$-differential privacy against user j, where d_{ij} is a distance function $d_{ij} : \mathcal{V} \times \mathcal{V} \to \mathbb{R}_+$ and $\epsilon : \mathbb{R}_+ \to \mathbb{R}_+$ is a decreasing function that converts distance d to a privacy level $\epsilon(d)$. Therefore, we need to design a mechanism that generates *accurate* responses $\{y_{ij}\}_{j \in \mathcal{V}}$ while satisfying *different privacy constraints for different recipients* based on the distance on the network. Specifically, accuracy is meant in the expected mean–squared error sense; the response y_{ij} should an accurate but private proxy of the private data u_i.

Additionally, we assume the existence of a trusted central authority. Users provide their

noiseless private data to this authority —already the case with modern social networks—, which executes a privacy–preserving mechanism, adds noise, and securely communicates the responses to each user. Then, differential privacy protects user's i private data from inference attacks by an adversarial user j while honest users locally run any post–processing such as a recommendation system. The assumption of a central authority can be relaxed by considering honest–but–curious users with only local secure communications.

3.1.2. Access Rights Scheme

Now, we describe a typical approach for handling privacy concerns in social network while highlighting its limitations and motivating the need for a more sophisticated privacy–aware approach. Figure 1a shows a synthetic network with 150 nodes, where the starred node wishes to share her sensitive information with the rest of the network. Privacy concerns can be handled by regulating access privileges. For example, friends of a user can access her data, whereas every other user cannot. Such a scheme has limitations. On one hand, users are coarsely partitioned to friends and strangers as depicted in Figure 1a; friends of the star-labeled user are colored white whereas strangers are colored black. On the other hand, the distance between two users can be more finely quantified by a real-valued function, and each user has access only to neighboring information. Although restricting access rights meets privacy concerns, computing global statistics on the network is impossible, limiting the global utility of the network. Indeed, any estimator of global quantities (mean value, histogram etc.) will to be biased. Therefore, the user may choose to collaborate, merge their local information, and damage any privacy guarantees. Figure 1b overcomes these limitations by defining a distance function $d : \mathcal{V} \times \mathcal{V} \to \mathbb{R}_+$ which quantifies the strength of the privacy concerns. In this case, users share privacy–aware versions of their profile with every member of the network.



(a) An access right scheme. (b) A distance-based scheme.

Figure 1: A synthetic network with 150 nodes and 1256 edges is shown. Each node represents a user of the network and each edge indicates a friendship. The user indicated with the star wishes to share her sensitive information with the rest of the network. Privacy concerns can be addressed by managing access privileges. Under an access right scheme (Figure 1a), only friends of the starred user (blue nodes) are granted access to the exact information, whereas any other member (red nodes) have no access. Such a scheme partitions users to only two groups; friends and strangers. Moreover, each user has access only to local information and cannot estimate the global state of the network. Therefore, any estimator constructed by the diamond user will be independed of the data of the starred user and, thus, biased. On the other hand, Figure 1b proposes an approach where users' privacy concerns scale with the distance from others. Friends (blue nodes) receive a less noisy versions of the private data, whereas strangers (red nodes) receive only heavily perturbed versions. Despite the increased noise, estimates of aggregate statistics are possible. However, coalitions might be encouraged and initial privacy guarantees can quickly degrade. For example, users within the circle can combine their estimates and infer the private data of the starred user.

3.1.3. Problem Statement

Under the modeling introduced in Subsection 3.1.1, we pose the problem of designing a mechanism that diffuses private data over a network as follows.

Problem 1. *Design a privacy–aware mechanism* $Q : \mathcal{U} \to \Delta\left(\mathcal{U}^N\right)$ *that privately releases user's* i *sensitive data* $u_i \in \mathcal{U}$ *over a social network. Specifically, design mechanism* Q *that generates* N *responses* $\{y_j\}_{j=1}^N$*, where* y_{ij} *is the securely communicated response to user* j*. Further, the mechanism* Q *needs to satisfy the following properties:*

- *Privacy: The mechanism must generate the response* y_{ij} *which preserves* $\epsilon(d_{ij})$*–differential private of data* u_i*.*
- *Utility: Response* y_{ij} *must be an accurate approximation of data* u_i*, i.e. for real–valued private data, it should minimize the expected squared–error*

$$\mathbb{E}_Q \|y_{ij} - u_i\|_2^2. \tag{3.1}$$

Specifically, whenever individual i shares her sensitive information to another individual j, she requires $\epsilon(d_{ij})$–differential privacy, where $\epsilon(\cdot) : \mathbb{R}_+ \to \mathbb{R}_+$ is a decreasing function that coverts a distance d to a privacy level $\epsilon(d)$. People residing close (w.r.t. a distance) to individual i receive a loose privacy constraint $\epsilon_{ij} \gg 1$, whereas strangers get noisier versions $\epsilon_{ij} \ll 1$.

Problem 1 admits a straightforward but unsatisfying approach. Let $y_j = u_i + V$, where $V \sim \text{Lap}\left(d_{ij}\right)$, independently for each user $j \in \mathcal{V}$. Subsequently, a group of users $j \in A \subseteq \mathcal{U}$ have the incentive to collaborate share their estimates $\{y_j\}_{j \in A}$ in order to derive a more accurate estimator y_A of u_i described by

$$y_A = \sum_{j \in A} w_j y_j. \tag{3.2}$$

Figure 1b depicts a group of users forming such a coalition. The possibly large group A resides far away from the user indicated by the star, $d_{ij} \gg 1, \forall j \in A$. Although each user j in the group A receives a highly noisy estimate of u_i, estimator y_A is more accurate. The composition theorem of differential privacy Dwork and Roth (2013) guarantees only $\left(\sum_{j\in A} \epsilon(d_{ij})\right)$-privacy which can be rather looser than each of the $\epsilon(d_{ij})$-privacy guarantees; larger values of ϵ imply less privacy.

Therefore, Problem 1 is subject to coalition attacks. Thus, we restate Problem 1 by requiring that any group A that exchanges their estimates $\{y_j\}_{j\in A}$ cannot produce a better estimator of u_i than the best estimator among the group y_{j^*}, where $j^* = \arg\min_{j\in A} d_{ij}$ is the user closest to user i. This problem can be stated as follows:

Problem 2. *Design a privacy-aware mechanism $Q : \mathcal{U} \to \Delta\left(\mathcal{U}^N\right)$ that releases a approximation of user's i sensitive data $u_i \in \mathcal{U}$ over a social network. Specifically, mechanism $\mathcal{M}$ generates N responses $\{y_j\}_{j=1}^N$ and securely communicates response y_j to user j. Mechanism Q needs to satisfy:*

- *Privacy: For any group of users $A \subseteq \mathcal{V}$, response $\{y_j\}_{j\in A}$ must be $\max_{j\in A} \epsilon(d_{ij})$–differential private.*
- *Utility: Response y_j must be an accurate approximation of the sensitive data u_i.*

3.2. Results

In this section, we derive a privacy–preserving mechanism that is resilient to coalitions. Subsection 3.2.1 derives the needed theoretical results and establishes that the accuracy of each estimate y_{ij} depends *only* on the distance d_{ij}. Moreover, algorithmic implementations of the composite mechanism Q should scale for vast social networks. Subsection 3.2.2 provides algorithmic implementations of the mechanism Q with complexity $O\left(\ln\left(\frac{\max_{i,j\in V} \epsilon(d_{ij})}{\min_{i,j\in V} \epsilon(d_{ij})}\right)\right)$.

3.2.1. Theoretical Result

For n–dimensional real–valued private data $u \in \mathbb{R}^n$, we derive a composite mechanism that generates the response y_{ij} that user j receives as an approximation of user's i private data u_i. This mechanism has the following two properties. First, the accuracy of the response y_{ij} depends solely on the distance d_{ij} between nodes i and j. Specifically, the expected squared–error $\mathbb{E}\|y_{ij} - u_i\|_2^2$ does not depend on any other parameters of the network (e.g. size, topology) or the rest of the responses $\{y_{ik}\}_{k\in\mathcal{V}\setminus\{j\}}$. Second, any group of users $A \subseteq \mathcal{V}$ that decides to collaborate and share their responses $\{y_{ij}\}_{j\in A}$ is *unstable*; i.e. the group does not learn more about u_i than what a member of the group already knows. Algorithmic aspects of the composite mechanism are deferred until Subsection 3.2.2.

Definition 7 introduces a continuous domain stochastic process $\{V_\epsilon\}_{\epsilon>0}$ which is used in Theorem 8 to define a composite privacy–preserving mechanism. Properties, sampling algorithms and the derivation of it are deferred for later.

Definition 7. *Define the stochastic process $\{V_\epsilon\}_{\epsilon>0}$ with the following properties:*

- *for $\epsilon > 0$, it is $d\mathbb{P}(V_\epsilon = v) \propto e^{-\epsilon\|v\|_2}$;*
- *the process is Markov; i.e. for any $0 < \epsilon_1 < \epsilon_2 < \epsilon_3$, it holds that $V_{\epsilon_1} \perp V_{\epsilon_3} | V_{\epsilon_2}$;*
- *for any $0 < \epsilon_1 < \epsilon_2$, with $\tau = \frac{\epsilon_2}{\epsilon_1} - 1$, it is*

$$\begin{aligned} d\mathbb{P}(V_{\epsilon_1} = v_1 | V_{\epsilon_2} = v_2) &\propto \delta(v_1 - v_2) \\ &+ \frac{(n+1)\epsilon_1^{1+\frac{n}{2}}\|v_1 - v_2\|_2^{1-\frac{n}{2}}}{(2\pi)^{\frac{n}{2}}} K_{\frac{n}{2}-1}(\epsilon_1\|v_1 - v_2\|_2)\tau \\ &+ O(\tau^2), \end{aligned}$$

where K is the modified Bessel function of the second kind.

Theorem 8. *Let $d_{ij} \in \mathbb{R}_+$ denote the distance between users i and j, and $u_i \in \mathbb{R}$ be the*

private data of user i. Consider the mechanism Q that generates the responses:

$$y_{ij} = u_i + V^{(i)}_{\epsilon(d_{ij})}, \tag{3.3}$$

where $\{V^{(i)}_\epsilon\}_{\epsilon>0}$ is a sample of a Markov stochastic process $\{V_\epsilon\}_{\epsilon>0}$. Then, mechanism Q provides a solution to Problem 2. In particular, it has the following properties:

- *The variance of response y_{ij} is $n\,(n+1)\,\epsilon(d_{ij})^{-2}$ and, thus, depends only on the distance between users i and j.*
- *For any subset of users $A \subseteq \mathcal{V}$, the mechanism that releases the responses $\{y_{ij}\}_{j\in A}$ is $\left(\max_{j\in\mathcal{A}} \epsilon(d_{ij})\right)$–differential private.*

The proof of Theorem 8 is presented in Appendix B. The main idea is to correlate the responses $\{y_{ij}\}_{j\in\mathcal{V}}$. For $n = 1$, the stochastic process $\{V_\epsilon\}_\epsilon$ has closed–form expressions, whereas, for $n > 1$, closed–form expressions are derived only for the infinitesimal increments $V_{\epsilon+d\epsilon} - V_\epsilon$. Nonetheless, we provide a sampling algorithm that allows for exact (in the sense that we do not use an approximation or discretization of the process) and efficient (in the algorithmic complexity sense) sampling of the process. Furthermore, our proof techniques are robust and can possibly be applied beyond the Laplace mechanism; for example, the K–norm mechanism Hardt and Talwar (2010) that appears in a different setting than the one considered here.

Figure 2 pictures two samples of the stochastic process $\{V_\epsilon\}_{\epsilon>0}$, for $n = 2$, in polar coordinates and shows that the process is a jump process; i.e., with high probability, the process is constant in small intervals. Figure 3 pictures two samples of the process in high dimensions. The process is again lazy, yet, the jumps are more often.

A major consequence of Theorem 8 is that mechanism Q does not incentivize coalitions. Specifically, consider a group of curious users $A \subseteq \mathcal{V}$ who wish to estimate u_i more accurately and, thus, collaborate and share their knowledge $\{y_{ij}\}_{j\in A}$. In practice, such a group can

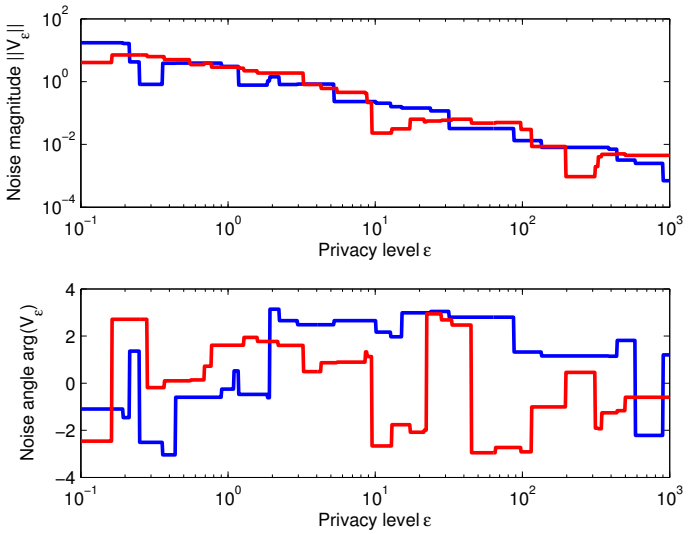


Figure 2: Two samples of the two-dimensional process which is the underlying object for diffusing private GPS coordinates over a network.

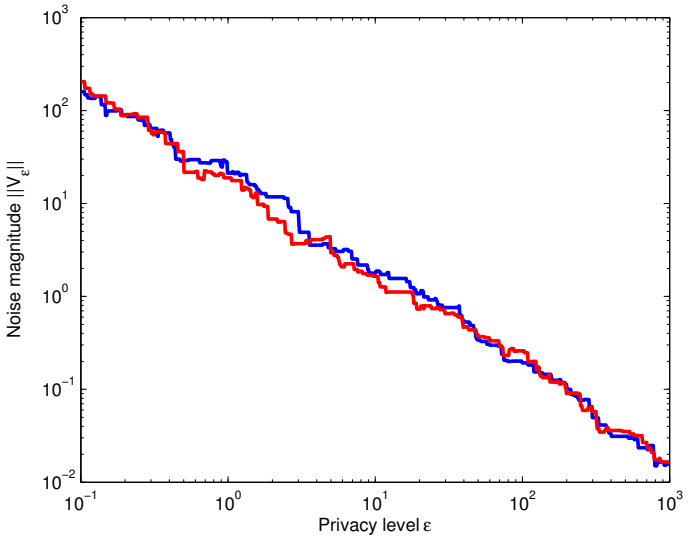


Figure 3: The ℓ_2-norm of two samples of the stochastic process $\{V_\epsilon\}_{\epsilon>0}$ in high-dimensions ($n = 20$) which can be used to diffuse private signals over networks, such as power consumption in smart grids.

be fake accounts of a single but distant (in the sense of d) user. Then, given this shared knowledge, the best estimator is:

$$\hat{u}_i = y_{ij^*}\big|_{j^* \in \arg\min_{j \in A} d_{ij}} . \tag{3.4}$$

Therefore, user j^* is not benefited by such a coalition and, thus, she has no incentive to participate in the coalition and share her information y_{ij^*}. In fact, Theorem 8 solves Problem 2 in the best possible way: the existence of multiple users asking for the same private data under different privacy levels *does not* require additional privacy–preserving noise.

3.2.2. Sampling Algorithm

Sampling from a continuous–domain stochastic process can often be performed only approximately. For example, consider the Brownian motion $\{B_t\}_{t\in[0,1]}$ which, for sampling purposes, requires storing an *infimum* of real values. Contrary to Brownian motion, the private process $\{V_\epsilon\}_{\epsilon>0}$ rarely changes value and is, thus, *lazy*. More formally, restricted to a sufficiently small interval $[\epsilon_1, \epsilon_2]$, the stochastic process $\{V_\epsilon\}_{\epsilon\in[\epsilon_1,\epsilon_2]}$ is constant with high probability. Furthermore, assuming the existence of an algorithm for computing the distance d_{ij}, the response y_{ij} can be generated during run-time. This property is crucial, since it circumvents the $O\left(N^2\right)$ memory requirements of a static implementation. Proposition 9 characterizes the distribution of the number of jumps in a bounded interval.

Proposition 9. *The number of jumps that the process $\{V_\epsilon\}_{\epsilon>0}$ performs in the interval $[\epsilon_1, \epsilon_2]$ is Poisson distributed with mean value $(n+1)\ln\left(\frac{\epsilon_2}{\epsilon_1}\right)$.*

$$\mathbb{P}\left(k \text{ jumps in } [\epsilon_1, \epsilon_2]\right) = \frac{x^k}{k!}e^{-x}, \tag{3.5}$$

where $x = (n+1)\ln\left(\frac{\epsilon_2}{\epsilon_1}\right)$.

Corollary 10. *Process $\{V_\epsilon\}_{\epsilon>0}$ performs $\mathbb{E}[k] = (n+1)\ln 2$ jumps (in expectation, with variance $Var[k] = (n+1)\ln 2$) for every doubling of the privacy level, i.e. in the interval*

$[\epsilon, 2\epsilon]$.

This laziness renders samples from the process highly–compressible. Indeed, given the locations $\{\epsilon^{(i)}\}_{i=1}^{k}$ of the jumps and the values* $\{V_{\epsilon_-^{(i)}}\}_{i=1}^{k}$ near those points a sample can be *exactly* reconstructed. The number k of jumps over a bounded interval $[\epsilon_1, \epsilon_2]$ is itself a random variable and captures the memory needs of our approach.

Furthermore, Proposition 9 suggests an efficient algorithm for directly sampling from the process $\{V_\epsilon\}_{\epsilon\in[\epsilon_1,\epsilon_2]}$, which we present in Algorithm 1. Algorithm 1 draws a sample $\{v_\epsilon\}_{\epsilon\in[\epsilon_1,\epsilon_2]}$ from the stochastic process V_ϵ over a bounded interval $\epsilon \in [\epsilon_1, \epsilon_2]$. This sample $\{v_\epsilon\}$ is the main object that performs diffusion of private data; whenever a user j requests user's i private data u_i residing d_{ij} away, the estimator $y_{ij} = u_i + v_{\epsilon(d_{ij})}$.

Algorithm 1 draws a sample of the stochastic process by sampling V_{ϵ_2}. Next, the algorithm proceeds towards smaller values of the privacy level ϵ by sampling the dormant time and the size of the jump. Specifically, the algorithm initializes a trace of the process by sampling from the Laplace mechanism. This is done is two steps; using the Gaussian distribution, the direction is drawn uniformly from the $n-1$–sphere and, then, the magnitude is drawn from the Gamma distribution. Then, the algorithm extends this trace backwards in ϵ by sampling for the location of the next jump. The logarithm of the positions where jumps occur define a Poisson process with rate $\lambda = n + 1$ and, thus, the length $\delta\epsilon = \ln \epsilon^{(i)} - \ln \epsilon^{(i+1)}$ of the interval until the next jump is exponentially distributed with density $\delta\epsilon \sim \lambda e^{-\lambda\,\delta\epsilon}$. Finally, conditioned on the event of a jump at $\epsilon^{(i)}$, the size $\delta v = V_{\epsilon_-^{(i)}} - V_{\epsilon_+^{(i)}}$ of the jump is Bessel–distributed with parameter $\frac{1}{\epsilon^{(i)}}$. The algorithm recycles until the level ϵ_1 is reached. Additionally, responses y_{ij} are generated upon request, and, thus, there is no excessive memory requirement $O(N^2)$ for storing all the responses $\{y_{ij}\}_{i,j\in\mathcal{V}}$. The number of iterations that Algorithm 1 performs is a random variable and is characterized by Proposition 9.

Typical single–dimensional ($n = 1$) private data are date of birth, salary, and health status.

*We use the notation $V_{\epsilon_-} = \lim_{\tau\uparrow\epsilon} V_\tau$ and $V_{\epsilon_+} = \lim_{\tau\downarrow\epsilon} V_\tau$.

For $n = 2$, our results are applicable to *geo–indistinguishability* Andrés et al. (2013) which is differential privacy for GPS locations and is experimentally illustrated in Subsection 3.3.1. Finally, the case $n \to \infty$ appeals to private signals that appear in filtering problems and smart grid applications.

For completeness, Table 1 presents the parameterization of the elementary distributions used by the proposed algorithms. We note that the Bessel distribution decays exponentially and has closed-form expressions for odd n. Nonetheless, it is a single-dimensional distribution and, thus, discritization and sampling through the inverse cumulative function is possible.

Algorithm 1 Sampling from the stochastic process V_ϵ over a bounded interval $\epsilon \in [\epsilon_1, \epsilon_2]$ can be performed both efficiently (with complexity $O\left(\ln\left(\frac{\epsilon_2}{\epsilon_1}\right)\right)$) and exactly (in the sense that we are not discretizing the interval or approximating the procssess).

Require: Dimension n; Privacy levels ϵ_1 and ϵ_2, such that $\epsilon_2 > \epsilon_1 > 0$.

function SamplePrivateProcessL2($n, \epsilon_1, \epsilon_2$)

$k \leftarrow 1$

$\epsilon^{(1)} \leftarrow \epsilon_2$

$r \sim \text{Gamma}\left(n, \frac{1}{\epsilon_2}\right)$

$v_1^{(1)}, \ldots, v_n^{(1)} \overset{\text{i.i.d.}}{\sim} \text{Gaussian}(0, 1)$

$v^{(1)} \leftarrow \frac{r}{\|v^{(1)}\|_2} v^{(1)}$

while $\epsilon(k) > \epsilon_1$ **do**

$\delta\epsilon \sim \text{Exponential}(n + 1)$

$\epsilon^{(k+1)} \leftarrow e^{-\delta\epsilon}\, \epsilon^{(k)}$

$r \sim \text{Bessel}\left(\frac{n}{2} - 1, \frac{1}{\epsilon^{(k+1)}}\right)$

$\delta v_1, \ldots, \delta v_n \overset{\text{i.i.d.}}{\sim} \text{Gaussian}(0, 1)$

$\delta v \leftarrow \frac{r}{\|\delta v\|_2} \delta v$

$v^{(k+1)} \leftarrow v^{(k)} + \delta v$

$k \leftarrow k + 1$

end while

Return $\{(\epsilon^{(i)}, v^{(i)}\}_{i=1}^{k}$

end function

3.3. Simulations

We present two application that depict diffusion of private data over a network. These example shows that bits of private information can be spread over the whole network, which

Distribution	**Param.**	**Supp.**	**Density**
Laplace	$\beta > 0$	$x \in \mathbb{R}$	$\frac{1}{2\beta} e^{-\frac{\lvert x \rvert}{\beta}}$
Exponential	$\lambda > 0$	$x \in \mathbb{R}_+$	$\lambda e^{-\lambda x}$
Gamma	$n \in \mathbb{N}$, $\beta > 0$	$x \in \mathbb{R}_+$	$\frac{1}{\Gamma(n)\beta^n} x^{n-1} e^{-\frac{x}{\beta}}$
Bessel	$n \in \mathbb{N}$, $\beta > 0$	$x \in \mathbb{R}_+$	$\frac{4}{\Gamma\left(\frac{n}{2}\right)(2\beta)^{\frac{n}{2}+1}} x^{\frac{n}{2}} K_{\frac{n}{2}-1}\left(\frac{x}{\beta}\right)$

Table 1: The distributions that are used by Algorithm 1. Sampling from these distributions can be performed using a uniform random variable and the quantile function.

allows users to estimate global quantities, such as epidemic spreading, while providing strong privacy guarantees.

3.3.1. Synthetic Data

We consider the synthetic network in Figure 5 with $N = |\mathcal{V}| = 150$ nodes and $|\mathcal{E}| = 1256$ edges, where edges are formed based on proximity. Each user $i \in \{1, \dots, N\}$ wishes to publish her vector–valued private data $u_i \in \mathbb{R}^2$, such as her GPS coordinates. For simplicity, we focus on a single user; our technique can be applied independently for each user. The distance d_{ij} between users i and j is captured by the shortest path length. We choose an exponential function $\epsilon(\cdot)$ that converts distances $d_{ij} \in \{1, \dots, 9\}$ to privacy levels $\epsilon(d_{ij}) \in [.5, 15]$. The function $\epsilon(\cdot)$ that converts distances d_{ij} to privacy levels $\epsilon(d_{ij})$ can be different for each agent. In fact, user i may require any privacy level against user j. In practice, these privacy levels can be manually chosen by each user or automatically generated by the system operator based on the structure of the network and preferences of the nodes. In any case, all privacy levels $\epsilon_{ij} = \epsilon(d_{ij})$ are assumed public knowledge.

Algorithm 1 is executed by user i for $n = 2$ and the norms of several traces are shown in Figure 4. For tight values of privacy level ($\epsilon \to 0$), large amounts of noises are added. In Figure 5, nodes are colored based on the accuracy $\|y_{ij} - u_i\|_2$ of the response y_{ij} they receive.

Although we have assumed the existence of a secure communication channel between any

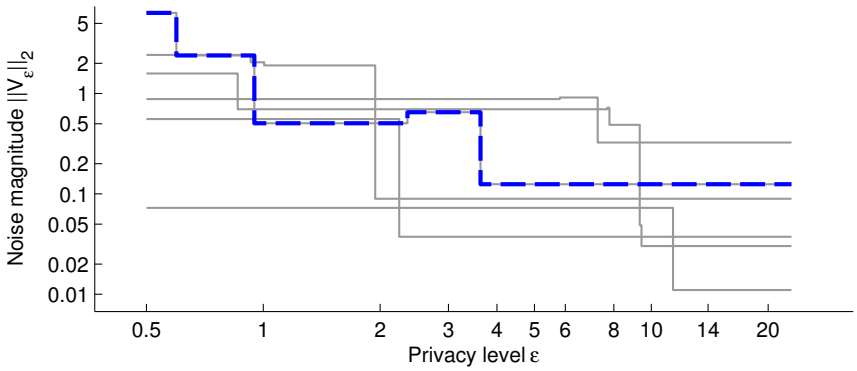


Figure 4: Agent i uses Algorithm 1 with $n = 2$ and generates a single sample of the stochastic process. For small values of privacy level, high noise values are more likely, whereas, for loose privacy levels ($\epsilon \to \infty$), the noise values decrease in magnitude. Despite the continuity of the domain $\epsilon \in [0, \infty)$, the process performs only a few jumps.

two users of the network and the existence of a central authority which computes the distances d_{ij}, an implementation that relaxes these assumptions is possible. Specifically, assuming only local communications between neighboring users, an *honest–but–curious* model, and knowledge of the privacy levels —which can be performed also in a decentralized manner— a distributed approach is possible. In such an implementation, user i sends to all her neighbors the signal $\{u_i + V_\epsilon\}_{\epsilon \in (0, \epsilon(1))}$. Then, each user j receives the signal $\{u + V_\epsilon\}_{\epsilon \in [0, \epsilon(d_{ij}))}$, trims it to $\{u + V_\epsilon\}_{\epsilon \in [0, \epsilon(d_{ij}+1))}$, and broadcasts it to her friends. An application of this variation is left for future work.

3.3.2. Real Dataset: Facebook

In this section, we present an application of diffusing sensitive data on a real network. Specifically, an *ego–network* was introduced by Leskovec and Mcauley (2012) and is a the sub–graph $G = (\mathcal{V} \cup \{\text{Alice}\}, \mathcal{E})$ of Facebook induced by a single user, Alice, and her friends $\mathcal{V}$. Figure 7 plots such an ego-network, where the bottom–left node is the user whose neighborhood is captured. The rest of the nodes represent Alice's friends, edges represent friendships between her friends, whereas, the edges between Alice and her friends are omitted for clarity. We assume that Alice's infection status is captured by a single bit $u \in \{0, 1\}$. Then, Alice wishes to share this information with her friends in a privacy–

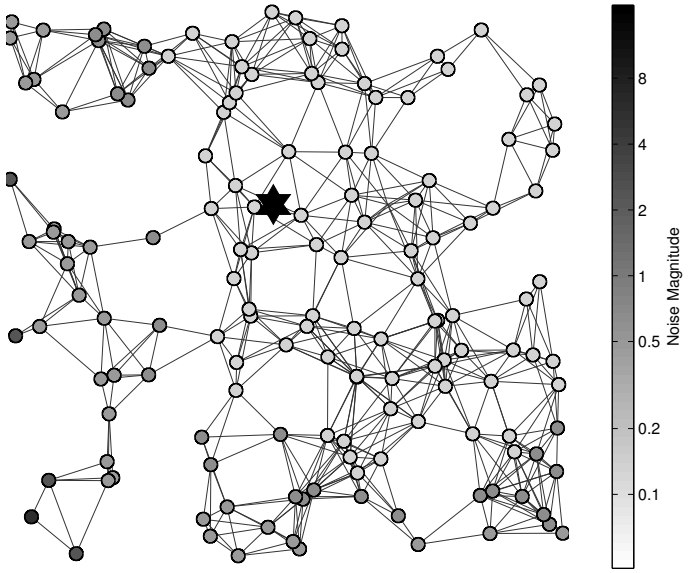


Figure 5: Each individual j gets the value $u_i + V_{\epsilon(d_{ij})}$, where u_i is the true sensitive data, d_{ij} is the number of hops between users i and j, and V_ϵ is the result of Algorithm 1.

preserving way.

For each friend $i \in \mathcal{V}$, the distance d_i is calculated by a central authority. Values $\{d_i\}_{i\in\mathcal{V}}$ are independent of the private data u, and can be computed without any privacy requirements. The strength of the friendship between Alice and friend i is quantified by the value of d_i and can be computed using methods from the social network literature such as the `score` functions suggested in Liben-Nowell and Kleinberg (2007). Here, distances d_i are evaluated according to Equation (3.6).

$$d_{ij} = \Gamma_{ii} + \Gamma_{jj} - 2\Gamma_{ij}, \tag{3.6}$$

where $\Gamma \in \mathbb{R}^{n\times n}$ is the pseudo–inverse of the Laplace matrix L of the network. Due to space limitations, we use the fact that our technique allows post–processing of the responses y_{ij} and, thus, is applicable for private bits.

Initially, Alice executes Algorithm 1 in order to generate a single sample $\{w_\epsilon : \epsilon \in [\underline{\epsilon}, \bar{\epsilon}]\}$ of the stochastic process $\{V_\epsilon : \epsilon > 0\}$, where $\underline{\epsilon}$ (resp. $\bar{\epsilon}$) is a lower (resp. upper) bound

of the quantity $\min_{i \in V} \epsilon(d_i)$ (resp. $\max_{i \in V} \epsilon(d_i)$). Function $\epsilon(\cdot) : \mathbb{R}_+ \to \mathbb{R}_+$ is a decreasing function which converts distances d_i to privacy levels $\epsilon_i = \epsilon(d_i)$. In this example, we chose $\epsilon(d) = exp(-3.3d + 4)$ which leads to privacy levels within $[.5, 15]$. In practice, any decreasing function resulting in any range of privacy levels can be used. The exact expression is considered a designer's choice that balances the users' privacy needs and the accuracy of network-wide statistics. Next, individual responses are generated during run-time. Whenever user i requests access to the sensitive data u, the response y_i is securely communicated to user i:

$$y_i = \Pi_{\{0,1\}}(u + w_{\epsilon(d_i)}), \tag{3.7}$$

where Π_S is the projection operator on the set S.

Figure 6 depicts two executions of Algorithm 1 with $n = 1$, whereas, Figure 7 plots the ego-network centered around Alice. In particular, Alice is shown on the bottom-left corner and each friend i is plotted at distance d_i from her. The blue and red circles mark the jumps of the stochastic process for the two samples w_ϵ^{blue} and w_ϵ^{red}. Counter-intuitively, friends i lying within two consecutive blue circles receive *exactly* the same response y_i although they are assigned different privacy levels $\epsilon(d_i)$. The paradox is settled by noticing that the boundary circles are random variables themselves. Therefore, users receiving identical responses have different confidence levels.

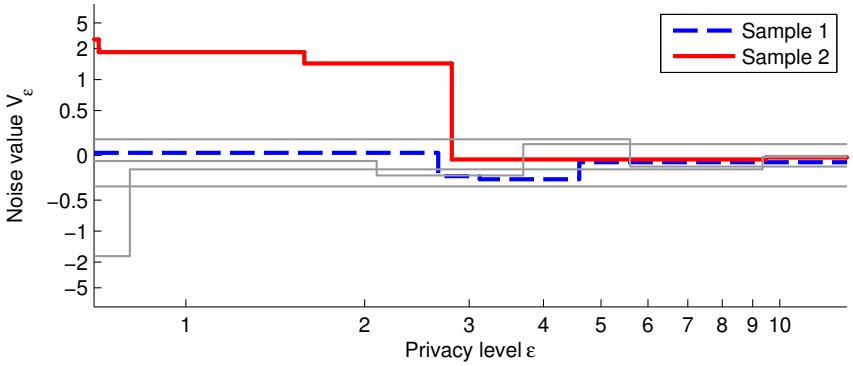


Figure 6: Two samples of the stochastic process generated by Algorithm 1. The samples are private information; a malicious user i can subtract the noise $w_{\epsilon(d_i)}$ from the received response y_i and exactly infer the private data u.

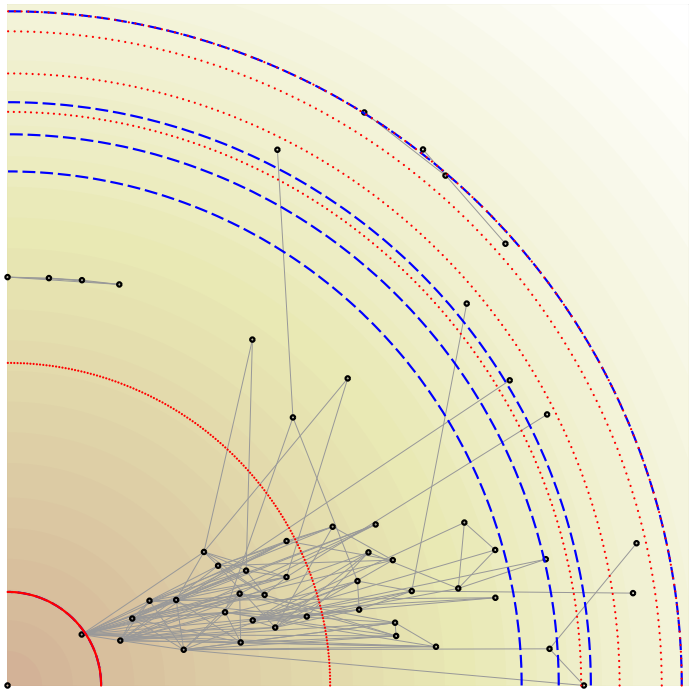


Figure 7: An ego-network is the part of the Facebook network that is visible from a fixed user A (ego), shown in the bottom-left corner of the plot. Each friend i is plotted at distance d_i. The locations of the jumps of the two samples shown in Figure 6 are depicted by the blue and red circles. Although users residing within consecutive circles receive identical responses y_i, they are assigned different privacy levels $\epsilon(d_i)$ and, thus, have different confidence levels.

CHAPTER 4: Gradual Release of Private Data

4.1. Motivation and Problem Formulation

Existing work on differential privacy assumes that the privacy level is determined prior to releasing any data and remains constant throughout the life of the privacy–preserving mechanism. However, for certain applications, the privacy level may need to be revised *after* data has been released, due to either users' need for improved accuracy or after owners' re–evaluation of the privacy concerns. One such application is trading of private data, where the owners re–evaluate their privacy concerns after monetary payments. Specifically, the end users initially access private data under ϵ_1 privacy guarantees and they later decide to *buy* more accurate data, relax privacy level to ϵ_2, and enjoy better accuracy. Furthermore, the need for more accurate responses may dictate a change in the privacy level. In particular, a database containing sensitive data is persistent over time; e.g. a database of health records contains the same patients with the same health history over several years. Future uses of the database may require better accuracy, especially, after a threat is suspected (e.g. virus spread, security breach). These two example applications share the same core questions.

Is it possible to release a preliminary response with ϵ_1–privacy guarantees and, later, release a more accurate and less private response with overall ϵ_2–privacy guarantees? How would this scenario compare to publishing a single response under ϵ_2–privacy guarantees? In fact, is the performance of the second response damaged by the preliminary one?

Composition theorems McSherry and Talwar (2007) provide a simple, but suboptimal, solution to gradually releasing sensitive data. Given an initial privacy level ϵ_1, a noisy, privacy–preserving response y_1 is generated. Later, the privacy level is relaxed to a new value ϵ_2, where $\epsilon_2 > \epsilon_1$, and a new response y_2 is published. For an overall privacy level of ϵ_2, it is sufficient for the second response y_2 to be $(\epsilon_2 - \epsilon_1)$–private, according to the composition theorem. Therefore, the accuracy of the second response deteriorates because of the initial

release y_1.

More sophisticated approaches can be defined such that the suboptimality of subsequent responses remains bounded. For instance, initially, we independently generate the responses $\{y_i\}_{i=-\infty}^{\infty}$, where y_i is 2^i–private. For an ϵ_1-private response, we release the sequence $\{y_i\}_{i=-\infty}^{\lfloor \log_2 \epsilon_1 \rfloor - 1}$. According to composition theorems, this sequence is ϵ_1–private and its accuracy is no worse than the accuracy of the last term $y_{\lfloor \log_2 \epsilon_1 \rfloor - 1}$, which is that of an $\frac{\epsilon_1}{2}$–private mechanism. As soon as the privacy level is relaxed from ϵ_1 to ϵ_2, more elements of the sequence are released. Such a setting partially handles the loss of accuracy in gradually relaxing the privacy level.

However, in this work, we derive an exact solution where there is *no* accuracy loss incurred. The solution to the problem formulated in this Chapter is closely related to the solution of Chapter 3 where the privacy levels at different times are viewed as different data users. Additionally, we extend the results to a broader class of existing privacy–preserving mechanisms beyond the identity queries considered in Chapter 3.

Section 4.2 states the result and provides an efficient algorithm, and Section 4.3 illustrates the applicability of them in Google's project RAPPOR.

4.2. Results

This work introduces the problem of gradually releasing sensitive data. Our results focus on the case of vector–valued sensitive data $u \in \mathbb{R}^n$ with an ℓ_1–norm adjacency relation. Our first result states that, for the one–dimensional ($n = 1$) identity query, there is an algorithm which relaxes privacy in two steps without sacrificing any accuracy. Although our technical treatment focuses on identity queries, our results are applicable to a broader family of queries; in particular, to a family of differentially private mechanisms that add Laplace noise. We also prove the *Markov property* for this algorithm and, thus, we can easily relax privacy in any number of steps; time complexity is linear in number of steps and memory complexity is constant. Gradually releasing sensitive data is performed by

sampling once from an underlying stochastic process. Although a different problem, the solution is provided by the same stochastic process derived in Chapter 3. Here we state the properties of this process in terms of the problem introduced, we extend our method for a broader class of mechanisms, and provide an efficient algorithm.

For a mechanism that performs gradual release of private data has the following properties, we require the following properties.

- **Privacy:** For any set of privacy levels $\{\epsilon_i\}_{i=1}^m$, the mechanism that responds with $\{Q_{\epsilon_i} u\}_{i=1}^m$ is $(\max_{i=1}^m \epsilon_i)$-private.
- **Accuracy:** For a fixed ϵ, the mechanism Q_ϵ is the optimal ϵ–private mechanism.

4.2.1. Theoretical Result

More formally, our main result derives a composite mechanism that gradually releases private data by relaxing the privacy level in an arbitrary number of steps.

Theorem 11 (Gradual Privacy as a Composite Mechanism)**.** *Let $\mathbb{R}^n$ be the space of private data equipped with an ℓ_1–norm adjacency relation. Consider m privacy levels $\{\epsilon_i\}_{i=1}^m$ such that $0 \leq \epsilon_1 \leq \cdots \leq \epsilon_m$ which successively relax the privacy level. Then, there exists a composite mechanism Q of the form*

$$Qu := (u + V_1, \ldots, u + V_m)\,, \tag{4.1}$$

such that:

1. *The restriction of the mechanism Q to the first j coordinates $(u + V_1, \ldots, u + V_j)$ is ϵ_j-private, for any $j \in \{1, \ldots, m\}$.*
2. *Each coordinate $j \in \{1, \ldots, m\}$ of the mechanism $u + V_j$ achieves the optimal mean-squared error $\mathbb{E}\|V_j\|_2^2 = 2n/\epsilon_j^2$.*

The mechanism that satisfies Theorem 11 has a closed–form expression which allows for

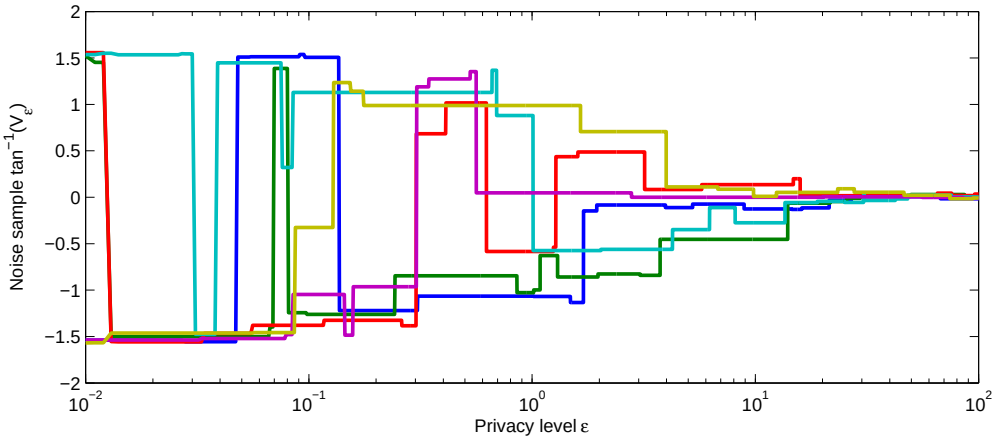


Figure 8: Gradual release of identity queries is achieved with the use of the stochastic process V_ϵ for $\epsilon \geq 0$, several samples of which are shown with different colors. In practice, a single sample of this process is drawn and, for a privacy level ϵ_0 and private data u, the noise version $u + V_{\epsilon_0}$ is published. For tight values of privacy ($\epsilon \to 0$), high values of noise ($\left|\tan^{-1} V_\epsilon\right| \to \frac{\pi}{2}$) are returned, whereas, almost zero samples ($V_\epsilon \to 0$) are returned for large privacy budgets ($\epsilon \to \infty$). The process V_ϵ is Markov; future samples depend only on the current value of the process which eases implementation. Furthermore, the process is lazy; the value of the process changes only a few times.

computationally lightweight implementation with $O(1)$ memory.

Theorem 11 is the result of the following two theorems. Theorem 12 handles a single round of privacy level relaxation and leads to Algorithm 2 and Theorem 13 proves a Markov property that allows the repetitive use of Theorem 12 for multiple rounds of privacy level relaxation.

Theorem 12. *Consider privacy levels ϵ_1 and ϵ_2 with $\epsilon_2 \geq \epsilon_1 > 0$, and mechanisms of the form:*

$$Q_1 u := u + V_1 \text{ and } Q_2 u := u + V_2, \text{ with } (V_1, V_2) \sim g, \tag{4.2}$$

for some probability density $g \in \Delta\left(\mathbb{R}^2\right)$. Then, for $g = l_{\epsilon_1,\epsilon_2}$ with:

$$l_{\epsilon_1,\epsilon_2}(x,y) = \frac{\epsilon_1^2}{2\epsilon_2} e^{-\epsilon_2|y|}\delta(x-y) + \frac{\epsilon_1(\epsilon_2^2-\epsilon_1^2)}{4\epsilon_2} e^{-\epsilon_1|x-y|-\epsilon_2|y|}, \tag{4.3}$$

where δ is the Dirac delta function, the following properties hold:

1. *The mechanism Q_1 is ϵ_1–private.*
2. *The mechanism Q_1 is optimal, i.e. Q_1 minimizes the mean–squared error $\mathbb{E}V_1^2$.*
3. *The mechanism (Q_1, Q_2) is ϵ_2–private.*
4. *The mechanism Q_2 is optimal, i.e. Q_2 minimizes the mean–squared error $\mathbb{E}V_2^2$.*

Proof. We straightforwardly verify that Distribution (4.3) has the aforementioned properties. □

Theorem 13. *Consider m privacy levels $\{\epsilon_i\}_{i=1}^m$ with $0 < \epsilon_1 < \cdots < \epsilon_m$ and mechanisms Q_i of the form:*

$$Q_i u = u + V_i, \text{ with } (V_1, \ldots, V_m) \sim g \in \Delta(\mathbb{R}^m). \tag{4.4}$$

Consider the distribution $g = l_{\epsilon_1,\ldots,\epsilon_m}$, with:

$$l_{\epsilon_1,\ldots,\epsilon_m}(v_1, \ldots, v_m) = l_{\epsilon_1}(v_1) \prod_{i=1}^{m-1} \frac{l_{\epsilon_i,\epsilon_{i+1}}(v_i, v_{i+1})}{l_{\epsilon_i}(v_i)}, \tag{4.5}$$

where $l_\epsilon(v) = \frac{\epsilon}{2} e^{-\epsilon|v|}$. Then, distribution $l_{\epsilon_1,\ldots,\epsilon_m}$ has the following properties:

1. *Each prefix mechanism $(Q_1, \ldots, Q_i)$ is ϵ_i–private, for $i \in \{1, \ldots, m\}$.*
2. *Each mechanism Q_i is the optimal ϵ_i-private mechanism, i.e. it minimizes the mean-squared error $\mathbb{E}V_i^2$.*

Theorem 11 performs gradual release of private data by releasing responses that approximate the identity query $q(u) = u$. In practice, however, the end–user is interested in more expressive queries q such as the mean value $\frac{1}{n}\sum_{i=1}^n u_i$ of a collection of private data $u_1, \ldots, u_n$ and solutions to optimization problems Han et al. (2014). Our results are directly applicable to a family of queries which are approximated by private mechanisms built around the Laplace

mechanism. Specifically, consider mechanisms based on the Laplace mechanism and have the form shown in Figure 9. The database of private data is initially pre-processed and, then, additive Laplace–distributed noise is used. The result is post-processed in order to maximize the accuracy of the response. Informally stated:

Corollary 14. *Let $(\mathcal{U}, d)$ be a metric space of sensitive data, $\mathcal{Y}$ be a set of responses, and $\epsilon > 0$ be a privacy level. Let*

- *$F : \mathcal{U} \to \Delta(\mathbb{R}^n)$ be a preprocessing step with sensitivity β. Function F is assumed to be invariant of ϵ, i.e. it does not change for different privacy levels,*
- *$\mathcal{L}_\epsilon : \mathbb{R}^n \to \Delta(\mathbb{R}^n)$ be the Laplace mechanism with parameter ϵ:*

$$\mathcal{L}_\epsilon u = u + V, \text{ where } V \sim e^{-\frac{\epsilon}{\beta}\|V\|_1}, \tag{4.6}$$

- *$G_\epsilon : \mathbb{R}^n \to \Delta(\mathcal{Y})$ be a post–processing step.*

Consider the ϵ–private mechanism

$$G_\epsilon \circ \mathcal{L}_\epsilon \circ F : \mathcal{U} \to \Delta(\mathcal{Y}). \tag{4.7}$$

Then, there exists a composite mechanism that performs gradual release of private data $u \in \mathcal{U}$.

In this case, the term *gradual release of private data* should be understood in the *scenario–replicating* sense; using Corollary 14 to relax privacy from ϵ_1 to ϵ_2, the resulting performance is the same as if the initial privacy level ϵ_1 had never occurred and the system was set at privacy level ϵ_2 from the beginning.

Finally, we provide another result that performs gradual release of private data under (ϵ, δ)–differential privacy. In that case, the underlying stochastic process is a reparameterization of the Brownian motion.

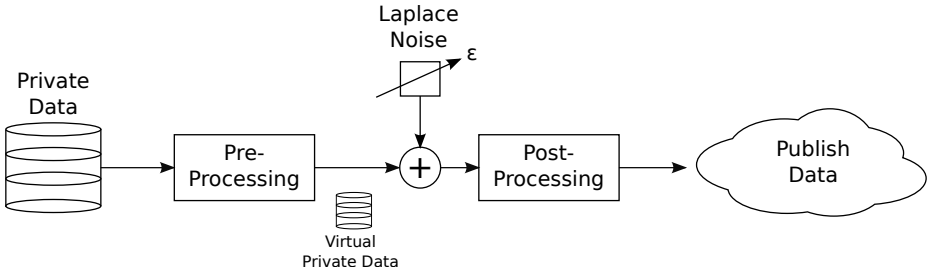


Figure 9: User 1 wants to share his sensitive data, such as his date of birth, in the a social network. Although, user 1 has no privacy concerns when sharing this information with his close friends 2 and 3, he has gradually increasing privacy issues for other members of the network. Specifically, a group A of distant users should not be able to collude and extract more information than what it is intended.

Theorem 15. *Let $\mathcal{U}$ be the space of privacy data, $q : \mathcal{U} \to \mathbb{R}^n$ be a query, and Δ be the ℓ_2-sensitivity for an adjacency relation $\mathcal{A}$. Then, consider the privacy levels $\{(\epsilon_t, \delta_t)\}_{t\in[0,\infty)}$ such that $\sigma(\epsilon_t, \delta_t)$ is a decreasing function of t. Then, the family of mechanisms Q_t:*

$$Q_t u := q(u) + V_t, \text{ where } V_t = \begin{bmatrix} B^{(1)}_{\sigma(\epsilon_t,\delta_t)} \\ \vdots \\ B^{(n)}_{\sigma(\epsilon_t,\delta_t)} \end{bmatrix} \text{ and } t \in (0,\infty), \tag{4.8}$$

where $V_t = \left[V_t^{(1)}, \ldots, V_t^{(n)}\right]$ and $V_t^{(i)} \overset{d}{=} B_{\sigma(\epsilon_t,\delta_t)}$ are independent samples of the Brownian motion B:

- ***Privacy:*** *For any $t > 0$, the mechanism that releases the signal $\{q(u) + V_\tau\}_{\tau\in(0,t]}$ is (ϵ_t, δ_t)-private.*
- ***Accuracy:*** *The mechanism Q_t that releases the random variable $q(u) + V_t$ is the Gaussian mechanism with $\sigma(\epsilon_t, \delta_t)$.*

We provide a short proof in Appendix C

4.2.2. Algorithm

In practice, gradual release of private data is achieved by sampling the stochastic process $\{V_\epsilon\}_{\epsilon>0}$ from left to right.

1. Draw a single sample $\{v_\epsilon\}_{\epsilon>0}$ from the stochastic process $\{V_\epsilon\}_{\epsilon>0}$.

2. Compute the signal $y_\epsilon = u + v_\epsilon$, $\epsilon > 0$.

3. For ϵ_1–privacy guarantees, release the random variable y_{ϵ_1}.

4. Once privacy level is relaxed from ϵ_1 to ϵ_2, where $\epsilon_2 \geq \epsilon_1$, release the random variable y_{ϵ_2}.

5. In order to relax privacy level in an arbitrarily many times, $\epsilon_1 \to \epsilon_2 \to \cdots \to \epsilon_m$, repeat the last step.

Specifically, Algorithm 2 is invoked repeatedly for each round of privacy level relaxation as depicted in Figure 10.

Algorithm 2 Sampling from Distribution (4.3) for the second noise sample $V_2 = y$ given the first noise sample $V_1 = x$ can be efficiently performed*.

Require: Privacy levels ϵ_1 and ϵ_2, such that $\epsilon_2 > \epsilon_1 > 0$, and noise sample x.

function RELAXPRIVACY($x, \epsilon_1, \epsilon_2$)

switch

case with probability $\frac{\epsilon_1}{\epsilon_2} e^{-(\epsilon_2-\epsilon_1)|x|}$:

return $y = x$.

case with probability $\frac{\epsilon_2-\epsilon_1}{2\epsilon_2}$:

draw $z \sim \begin{cases} e^{(\epsilon_1+\epsilon_2)z}, & \text{for } z \leq 0 \\ 0, & \text{otherwise.} \end{cases}$

return $y = \text{sgn}(x)z$.

case with probability $\frac{\epsilon_1+\epsilon_2}{2\epsilon_2}\left(1 - e^{-(\epsilon_2-\epsilon_1)|x|}\right)$:

draw $z \sim \begin{cases} e^{-(\epsilon_2-\epsilon_1)z}, & \text{for } 0 \leq z \leq |x| \\ 0, & \text{otherwise.} \end{cases}$

$y = \text{sgn}(x)z$.

case with probability $\frac{\epsilon_2-\epsilon_1}{2\epsilon_2} e^{-(\epsilon_2-\epsilon_1)|x|}$:

draw $z \sim \begin{cases} e^{-(\epsilon_1+\epsilon_2)z}, & \text{for } z \geq |x| \\ 0, & \text{otherwise.} \end{cases}$

return $y = \text{sgn}(x)z$.

end switch

end function

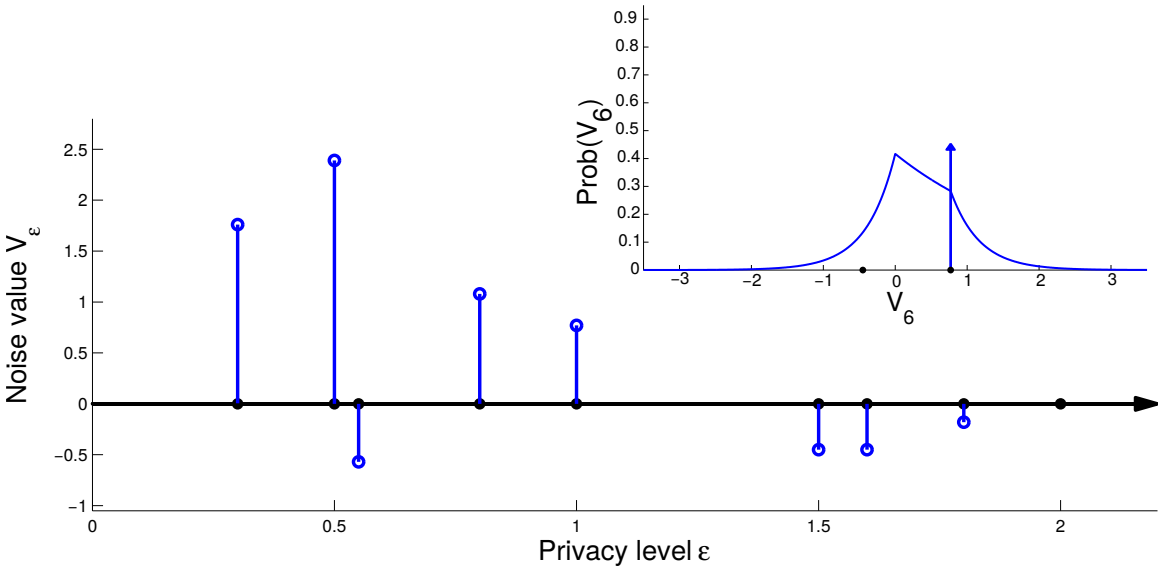


Figure 10: Privacy level can be repeatedly relaxed. For each round of relaxation $\epsilon_i \to \epsilon_{i+1}$, the distribution of the next noise sample V_{i+1} depends only on the last noise sample V_i. Past noise samples $\{V_j\}_{j<i}$ can be discarded from memory, thus, there is no complexity incurred from repeatedly relaxing privacy level.

4.3. Application

Examples of existing such privacy–aware mechanisms can be found in e.g. smart grids Koufogiannis et al. (2014) and user's reports Erlingsson et al. (2014). On the other hand, our results do not address the gradual release of private through mechanisms that do not fulfill this assumption, such as privately solving optimization problems with stochastic gradient descent Han et al. (2014).

In particular, Google's RAPPOR Erlingsson et al. (2014) is a mechanism that collects private data from multiple users for "crowdsourcing statistics" and can be expressed in terms of the Laplace mechanism. RAPPOR collects personal information from users such as the software features they use and the URLs they visited, and provides statistics of this information over a population of users. Algorithmically, a Bloom filter B of size k —which is a bank of k hashing function in parallel— is applied to each user's private data u:

$$B : \mathcal{U} \to \{0,1\}^k, \quad y = [y_1, \ldots, y_k] = B(u), \tag{4.9}$$

where $\mathcal{U}$ is the space of private data, in particular, the set of all strings. Next, each bit y_i is perturbed with probability f and the result is memoised:

$$f : \{0,1\}^k \to \{0,1\}^k, \quad z = [z_1, \ldots, z_k] = f(y), \text{ where } z_i = \begin{cases} 0, & \text{w.p. } \frac{1}{2}\alpha, \\ 1, & \text{w.p. } \frac{1}{2}\alpha, \\ y_i, & \text{w.p. } 1 - c_1, \end{cases} \tag{4.10}$$

where "w.p." stands for "with probability" and $\alpha \in [0,1]$ is a parameter. Finally, RAPPOR applies another perturbation each time a report is communicated to the server. This perturbation is equivalent to the map (4.10) but differently parametrized:

$$g : \{0,1\}^k \to \{0,1\}^k, \quad w = [w_1, \ldots, w_k] = f(z), \text{ where } \mathbb{P}(w_i = 1) = \begin{cases} \beta, & \text{if } z_i = 1, \\ \gamma, & \text{if } z_i = 0, \end{cases} \tag{4.11}$$

where $\beta, \gamma \in [0,1]$ are parameters. RAPPOR's differential privacy guarantees relax (increased ϵ) for small values of α and γ, and large values of β.

An important limitation of RAPPOR is that parameters α, β, and γ are forever fixed. However, there are reasons that require the ability to update these values in a way that the privacy is relaxed and the accuracy is increased:

- Due to the non-trivial algorithm of decoding the reports, a tight accuracy-analysis is not possible. Instead, the accuracy of the system is evaluated once the system is bootstrapped.[†] Our results make it possible to initialize the parameters with tight values $\alpha \to 1$, $\beta \to .5$, $\gamma \to .5$, and subsequently relax the parameters until a desired accuracy is achieved.

- Once a process or URL is suspected as malicious, the server would be interested

[†]Even in that case, estimating the actual accuracy can be challenging since it should be performed in a differential private way.

in relaxing the privacy level and performing more accurate analysis of the potential threat. Once such a threat is identified, our result allows users to gradually relax their privacy parameters and the server can more confidently explore the potential threat.

In order to apply Theorem 11 to RAPPOR, we express the randomized maps (4.10) and (4.11) using the Laplace mechanism. Specifically, consider the functions $\bar{f}$ and $\bar{g}$ that add Laplace noise and project the result to $\{0,1\}$:

$$\bar{f}(\psi) = \left[\psi + V_f > \frac{1}{2}\right], \quad \text{where } V_f \sim \text{Lap}\left(\frac{1}{-2\ln\alpha}\right), \tag{4.12}$$

$$\bar{g}(\zeta) = \left[\zeta + V_g > \frac{\ln(2\gamma)}{\ln\left(4\gamma(1-\beta)\right)}\right], \quad \text{where } V_g \sim \text{Lap}\left(\frac{1}{-\ln\left(4\beta(1-\gamma)\right)}\right), \tag{4.13}$$

where $\psi, \zeta \in \{0,1\}$, $\text{Lap}(b)$ is the Laplace distribution with parameter b, and the bracket symbol $[\cdot] \in \{0,1\}$ is 1 if, and only if, the inside expression is true. Note that functions $\bar{f}$ and $\bar{g}$ have the structure of Figure 9. Moreover, it can be shown that $\bar{f}$ and $\bar{g}$ applied component-wise to y and z are reformulations of the maps f and g. Therefore, privacy level relaxation is achieved by sampling noises V_f and V_g as suggested by our results. Note that, due to the Markov property, past privacy levels need not be stored in memory; only the currently applicable privacy level is retained.

CHAPTER 5: Privacy of Current State

5.1. Motivation

For *time–varying private data*, we may wish to only protect the privacy of the *current* value of the private data. In such settings, we need to block an inference attack on the sensitive data at the current time and not necessarily the whole trajectory as explored by Le Ny and Pappas (2014). Specifically, *at each time step and given the responses published so far, we need to defend against an adversary that attempts to infer the current value of private data, i.e. the current state of the system.*

Specifically, existing work in differential privacy assumes that the private data is given and fixed in time and proposes privacy–mechanisms for a wide spectrum of applications. Even for e.g. private signals, the proposed mechanisms protect the privace of the signal as a whole. As an implication, in dynamical phenomena, these works provide only *static* privacy guarantees, i.e. the mechanism that maps private data to the responses is differentially private. In practice, however, privacy needs may vary over time in one or more of the following aspects: (i) the private data itself may change over time, (ii) additional responses may be published, or (iii) the strength of privacy may be either revised at a later time. More concretely, consider an individual using a location–based service and, thus, reporting her GPS location. Such an individual may wish to protect her *current* location, while not worrying about revealing her past locations. From a different point of view, an adversary may wish to track the state of a dynamical system and decide when to deploy an attack. In each case, we need to provide privacy guarantees that explicitly protect the *current* state of the system. Motivated by such applications, this work introduces time–varying privacy guarantees; at each time, the mechanism publishes additional information and the private data evolves. Then, we wish to design a privacy–preserving mechanism such that an adversary who observes the so–far responses of the mechanism cannot confidently infer the *current* private data. In Chapter 4 we explored the case where only the strength of the

required privacy varies over time.

Our work deviates from the literature by considering *time–varying* differentially private guarantees. For an underlying dynamical system, we formulate and solve the problem of designing a mechanism that provides the following privacy guarantees. At each time step, an adversary that has observed the outputs of the mechanism so far cannot confidently infer the current state of the system. The time–varying sense of a privacy statement stems from the fact that the mechanism publishes new outputs with every step, thus, offering to the adversary additional knowledge. Moreover, the private data that needs to be protected is the current state which changes over time. On a technical note, we also allow the privacy level, i.e. the strength of the privacy guarantees, at each time step to vary as well; either increase or decrease at each time step. Our contributions are both conceptual and technical. Conceptually, we extend differential privacy for the case where the private is not a fixed quantity. Also, the proposed mechanism overcomes the problem of *depletion of privacy budget* by changing the private data itself and, thus, allows for infinite horizons while maintaining meaningful privacy guarantees and accuracy of the responses. Additionally, contrary to existing privacy–preserving mechanisms that inject noise only in the published responses, our mechanism consists of two noise sources: aside from corrupting the published responses with noise, the mechanism perturbs the private data itself. Regarding our technical contributions, we design a Gaussian–based privacy–preserving mechanism for a linear system that provides (ϵ, δ)–differential privacy. Additionally, for scalar system under a ϵ–differential privacy, we provide an efficient privacy–preserving mechanism that, at each time, publishes the most accurate but private approximation of the current state. Both mechanisms consist of two parts: the sensor part which misreports the current private data by publishing only noisy versions of it and is typically used in differential privacy literature, and the controller part which injects noise to the system and corrupts the private data itself.

5.2. Problem Formulation

5.2.1. Time–varying Private Data

As a motivating example, we consider a swarm of mobile agents collaboratively monitoring a quantity of interest —e.g. a target's position or a temperature field— and publishing an estimate of this quantity. Additionally, the agents themselves do not want to be tracked and, thus, have privacy needs for their *current* state —e.g. an adversary may try to localize and attack them. Since the agents' positions may be inferred from the published responses, we wish to design a privacy–preserving mechanism that publishes accurate information while guaranteeing the agents' privacy. Another example, considers a vehicle traveling on a highway segment and reporting its position for traffic–monitoring purposes. However, due to privacy concerns, the vehicle does not wish to be accurately localized on the highway at any time.

A key observation, to be exploited later, is that if the privacy requirements cannot be satisfied by solely perturbing the published responses, the agent noisily perturbs its private data. This observation deviates from the assumptions in the differential privacy literature where the private data are assumed given and fixed over time and the mechanism cannot tamper with them. In practice, although some private data such as health records cannot be altered, in several scenarios, private data including sensor locations, leadership tokens, and a dynamical system's state can be updated by a mechanism.

We introduce our problem in its general form and, later, we will focus on linear instances of it. Formally, we consider a dynamical system with state x_t and open–loop dynamics $x_{t+1} = f(x_t, u_t)$. For each time $t \in \{1, \dots, T\}$, we wish to publish the observations $y_t = g(x_t)$. However, due to privacy concerns, at time t, we wish to protect the privacy of the current state x_t by appropriately injecting noise. Importantly, the privacy constraints are time–varying; the data that needs to remain private is not always the same but it evolves with time. Moreover, the adversary's knowledge changes as additional observations are

published and, thus, past noisy observations potentially can harm the privacy of the current private data. To this end, we wish to design a privacy–preserving mechanism such that, at time t, the mechanism that maps the current state x_t to the so–far published observations $\{y_s : s \in 1:t\}$ is ϵ_t–differentially private. The sequence of privacy levels $[\epsilon_t]_{t=1}^T$ is assumed to be given.

Contrary to existing privacy–preserving mechanisms that only perturb the published responses, the approach proposed in this paper considers mechanisms that inject noise both in the sensor and the controller, as depicted in Figure 11.

i. *Sensor noise:* instead of the exact measurement y_t, the mechanism only publishes noisy versions of it $\hat{y}_t \sim \mathcal{G}(x_t)$; for example, $\hat{y}_t = y_t + V_t$, where V_t is suitable privacy-preserving noise. Intuitively, noise V_t protects the current state x_t from an adversary that knows the current observation $\hat{y}_t$. This type of noise is similar to the noise added by existing privacy–preserving mechanisms.

ii. *Controller noise:* the mechanism injects noise to the input of the system, $u_t \sim \mathcal{H}(u_t^{(0)})$, where $\mathcal{H}$ is a suitable privacy-preserving mechanism; for example $u_t = u_t^{(0)} + W_t$, where $u_t^{(0)}$ is an external control input —for simplicity we assume $u^{(0)} \equiv 0$— and W_t is appropriate noise. In words, if past observations $\hat{y}_1, \ldots, \hat{y}_{t-1}$ can be used to accurately infer the next state x_t, then, the injected noise perturbs the system's state itself to enforce privacy.

Regarding performance of the system, we wish to minimize the amount of injected noise; increased sensor noise V_t renders the measurements $\hat{y}_t$ uninformative, whereas increased noise W_t changes the control input from the nominal one and, thus, degrades the performance of the plant.

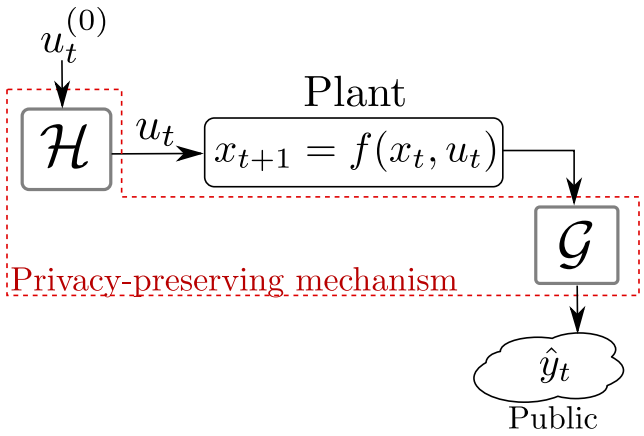


Figure 11: We wish to design a privacy–preserving mechanism $(\mathcal{H}, \mathcal{G})$ such that, at time t and given the published observations $\{\hat{y}_i\}_{i=1}^t$, the current state x_t is ϵ_t–differentially private.

5.2.2. Problem Statement

Finally, we concretely formulate the problem of designing a mechanism that, at each time, guarantees the privacy of the current state of a dynamical system. In this work, we consider linear dynamical systems

$$x_{t+1} = A_t\, x_t + B_t\, u_t, \tag{5.1}$$

$$y_t = C_t\, x_t \tag{5.2}$$

Given a nominal input $u_t^{(0)}$, we wish to design a privacy-preserving mechanism $(\mathcal{H}, \mathcal{G})$ that sets $u_t = \mathcal{H}(u_t^{(0)})$ and $\hat{y}_t = \mathcal{G}(y_t)$ as depicted in Figure 11. Specfically, this mechanism is formulated in Problem 3.

Problem 3. *Given a sequence of privacy levels $[(\epsilon_t, \delta_t)]_{t=1}^T$, design a mechanism $(\mathcal{H}, \mathcal{G})$ such that*

- *Privacy: at time t, state x_t is (ϵ_t, δ_t)-private, i.e.*

$$\mathbb{P}(\hat{y}_1, \ldots, \hat{y}_t | x_t) \leq e^{\epsilon_t}\, \mathbb{P}(\hat{y}_1, \ldots, \hat{y}_t | x_t') + \delta_t, \tag{5.3}$$

  for adjacent $(x_t, x_t') \in \mathcal{A}$;

- *Performance: the amount of injected noise is minimized*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[C_1(0,u_t)\right]+\mathbb{E}\left[C_2(y_t,\hat{y}_t)\right], \tag{5.4}$$

where we assume that $u_t^{(0)}=0$ *and* C_1 *and* C_2 *are cost functions that penalize excessive noise.*

5.3. Results

We now solve two instances of Problem 3. Subsection 5.3.1 considers a linear system under (ϵ,δ)–differentially private guarantees and, next, shows that our technique allows for time–varying privacy levels. Subsection 5.3.2 considers a scalar linear time–varying system under ϵ–differential privacy and provides a simple privacy–preserving mechanism that allows infinite horizon.

5.3.1. Linear System under (ϵ,δ)–Differential Privacy

We now design a Gaussian–based privacy–preserving mechanism that solves Problem 3 for a linear system in an LQG–like setting. Specifically, we consider the following linear, for simplicity time invariant, system.

$$x_{t+1}=A\,x_t+B\,u_t,\quad\text{and}\quad y_t=C\,x_t. \tag{5.5}$$

We assume that the system parameters A, B, and C are publicly known, that the system starts at $t=0$, but the first observation published is y_1, and that (ϵ,δ) is a given privacy level. Our results remain applicable for time–varying privacy levels, i.e. a given sequence $[(\epsilon_t,\delta_t)_{t=1}^{T}$ of privacy levels, where T is a time horizon. Finally, we consider the adjacency

relation $(x_t, x'_t) \in \mathcal{A} \Leftrightarrow \|x_t - x'_t\|_2 \leq 1$ and the quadratic cost

$$C_T = \frac{1}{T}\left[\sum_{t=0}^{T-1} \mathbb{E}\left(u_t - u^{(0)}\right)^T R \left(u_t - u^{(0)}\right) + \right. \tag{5.6}$$

$$\left. \sum_{t=1}^{T} \mathbb{E}\left(\hat{y}_t - y_t\right)^T Q \left(\hat{y}_t - y_t\right)\right], \tag{5.7}$$

where we assume $R \succeq 0$ and $Q \succeq 0$ are positive semi–definite matrices that penalize control and output noise, respectively. Additionally, we assume that the nominal input $u_t^{(0)}$ is publicly known and, thus, we can ignore it by assuming $u_t^{(0)} = 0$. Since the input signal may be computed based on public information or be inferred by past executions of the system, we cannot argue about the privacy of the nominal input. Thus, following the dogma of differential privacy for a powerful adversary, we assume that this signal is publicly known.

In order to design a mechanism that, at time t, guarantees (ϵ, δ)–privacy of the current state x_t with respect to the adjacency relation $\mathcal{A}$, we design a privacy–preserving mechanism of the form

$$u_t = u_t^{(0)} + W_t = W_t, \quad \text{and} \quad \hat{y} = y_t + V_t. \tag{5.8}$$

Then, Problem 3 is stated as in Problem 4.

Problem 4. *Design the stochastic processes $[W_t]_{t=1}^T$ and $[V_t]_{t=1}^T$ such that the privacy–preserving mechanism that inputs $u_t = W_t$ and publishes $\hat{y}_t = y_t + V_t$ satisfies the following properties.*

- *At time t, the current state x_t is (ϵ, δ)–differentially private.*
- *The quadratic cost C_T is minimized; i.e. the processes W and V are not unnecessarily noisy.*

For this problem, we consider only zero–mean Gaussian–based privacy–preserving mecha-

nisms of the form shown in Equation (5.8). Specifically, we design the covariance matrix

$$\begin{pmatrix} E_t \\ W_t \\ V_{t+1} \end{pmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \mathbf{\Sigma}_t & 0 & \mathbf{X}_t \\ 0 & \mathbf{W}_t & \mathbf{Y}_t \\ \mathbf{X}_t^T & \mathbf{Y}_t^T & \mathbf{Z}_t \end{bmatrix} \right), \tag{5.9}$$

where $E_t = \hat{x}_t - x_t$ and $\hat{x}_t$ is the least–squares estimator of x_t, given the responses $\{\hat{y}_i\}_{i=1}^t$.

In the structure of the correlation matrix in Equation (5.9), we allow for correlation between the input and the output noise. However, we chose not to allow for any correlation between the input noise and the error of the least–squares estimator. These properties are similar to the mechanism presented in Subsection 5.3.2 for a scalar system under ϵ–differential privacy.

The covariance matrix in Equation (5.9) is derived from the following convex optimization problem.

$$\begin{aligned} \underset{\{\mathbf{\Sigma}_t, \mathbf{W}_t, \mathbf{Z}_t, \mathbf{X}_t, \mathbf{Y}_t\}_{t=1}^T}{\text{minimize}} \quad & \sum_{t=0}^{T-1} \Big[\operatorname{tr}(R\,\mathbf{W}_t) + \operatorname{tr}(Q\,\mathbf{Z}_t) \Big] \\ \text{s.t.} \quad & \begin{bmatrix} \mathbf{\Sigma}_t & 0 & \mathbf{X}_t \\ 0 & \mathbf{W}_t & \mathbf{Y}_t \\ \mathbf{X}_t^T & \mathbf{Y}_t^T & \mathbf{Z}_t \end{bmatrix} \succeq 0, \\ & \begin{bmatrix} M_t - \mathbf{\Sigma}_{t+1} & N_t + M_t C^T \\ * & C M_t C^T + Z_t + \operatorname{sym}(C\,N_t) \end{bmatrix} \succeq 0 \\ & \mathbf{\Sigma}_t \succeq \kappa^{-2}(\epsilon, \delta)\, I, \quad \forall t, \end{aligned} \tag{5.10}$$

where matrices M_t and N_t are linear functions of the variables defined in the proof of Theorem 16. The first constraint requires that the covariance block–matrix is well-defined, whereas the second constraint recursively couples the covariance matrices across different times. Lastly, the third inequality enforces the privacy constraint.

The following result proves that, for a feasible solution of program in Equation (5.10), the current state x_t is (ϵ, δ)–private.

Theorem 16. *Consider the linear system* (5.5) *with* $u_t = W_t$ *and* $\hat{y}_t = y_t + V_t$ *as defined in* (5.8) *and a privacy level* $[(\epsilon, \delta)]_{t=1}^{T}$. *If the covariance matrix satisfies the constraints of the optimization problem in* (5.10)*, then, at time* t *and given the observations* $[\hat{y}_i]_{i=1}^{t}$*, the current state* x_t *of the system is* (ϵ, δ)*–differentially private.*

The proof of this result is included in the Appendix. The derivation of the result follows the steps of Kalman estimation, in particular, as in Tanaka et al. (2014) and, then, invoking the Gaussian mechanism from differential privacy. However, here we allow for correlation between the control noise, the sensor noise, and the estimation error and, thus, the exact expression is different. Specifically, we guarantee (ϵ, δ)–differential privacy for x_t at time t, if the least–squares estimator $\hat{x}_t$ can be written in the form of a Gaussian mechanism

$$\hat{x}_t = x_t + E_t, \tag{5.11}$$

where $\mathrm{Var}(E_t) \geq \kappa^2(\epsilon, \delta)\, I$.

The following result provides sufficient conditions for the feasibility of the optimization problem.

Proposition 17. *If the matrix* $[A; B]$ *has full row rank, then, the problem in Equation 5.10 is feasible.*

Sampling for the privacy–preserving noises $[W_t]_{t=0}^{T-1}$ and $[V_t]_{t=1}^{T}$ can be done as follows.

- The sensor part initializes the Kalman estimator by choosing $E_0 \sim \mathcal{N}(0, \mathbf{\Sigma}_0)$.
- At each time t, the controller part draws $W_t \sim \mathcal{N}(0, \mathbf{W}_t)$.
- At time t, the sensor part measures the state x_{t+1}, infers W_t and E_t, draws V_{t+1} conditioned on W_t and E_t, and publishes the response $\hat{y}_{t+1} = C\, x_{t+1} + V_{t+1}$.

Finally, we highlight that our technique allows for different privacy levels at each time step t, which captures the scenario where the privacy of part of the trajectory needs to be better protected. Specifically, given a sequence of privacy levels $[(\epsilon_t, \delta_t)]_{t=1}^{T}$, we can replace the last constraint in Equation (5.10) with the time–dependent expression

$$\mathbf{\Sigma}_t \succeq \kappa^{-2}(\epsilon_t, \delta_t)\, I, \quad \forall t \in \{1, \dots, T\}. \tag{5.12}$$

5.3.2. Scalar System under ϵ–Differential Privacy

In this section, we consider a scalar system and ϵ–differential privacy and we provide a simple Laplace–based privacy–preserving mechanism that, at time t, protects the current state x_t with a privacy level ϵ_t. Specifically, we consider a noiseless scalar system with state $x_t \in \mathbb{R}$ and publicly known dynamics

$$x_{t+1} = a_t\, x_t + u_t \quad \text{and} \quad y_t = x_t, \tag{5.13}$$

a sequence of privacy levels $[\epsilon_t]_{t=1}^{T}$, where $T \in \mathbb{N} \cup \{\infty\}$ is a, possibly infinite, time horizon, and the adjacency relation $\mathcal{A}$ defined as

$$(x_t, x_t') \in \mathcal{A} \Leftrightarrow |x_t - x_t'| \leq 1. \tag{5.14}$$

At time t, the value of ϵ_t captures the strength of the privacy guarantees. Importantly, we do not make any assumptions on the monotonicity of the sequence of privacy levels and, thus, we allow for both privacy relaxation and tightening over time. For constant private data $x_t = x$, $\forall t$, the problem of *relaxing* privacy (increasing sequences of ϵ_t) has been explored in earlier work (Koufogiannis et al. (2016)) but in the case of fixed private data, whereas, privacy tightening is conceptually impossible; once a response is published it is impossible to recall it. In our setting, we overcome this limitation by allowing for the privacy–preserving mechanism to noisily change the private data itself. Specifically, given

the system in Equation (5.13), we consider a mechanism of the form

$$u_t = W_t \quad \text{and} \quad \hat{y}_t = y_t + V_t, \tag{5.15}$$

where $[W_t]_{t=1}^T$ and $[V_t]_{t=1}^T$ are appropriate privacy–preserving stochastic processes. As mentioned earlier, the input noise W_t changed the private data and, thus, at time $t+1$, we need to protect the new private data $a_t\, x_t + W_t$. Contrary to W_t which becomes part of the private data, the output noise V_t is logistic; essentially the mechanism misreports its state. Regarding accuracy, we consider a cost that penalizes inaccurate published data

$$C_T = \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\,(\hat{y}_t - y_t)^2 = \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\, V_t^2. \tag{5.16}$$

Then, Problem 3 takes the more specific form of Problem 5.

Problem 5. *Design the stochastic processes $[W_t]_{t=1}^T$ and $[V_t]_{t=1}^T$ such that*

- *for each time t and given the current state x_t, the mechanism that publishes $\{\hat{y}_i\}_{i=1}^t$ is ϵ_t–differentially private; and*
- *the published responses $\hat{y}_t$ accurately approximate the desired output x_t; i.e. minimizes the cost C_T.*

Theorem 18 solves Problem 5 and hints to an efficient algorithm that draws a sample from the stochastic processes W_t and V_t. In order to state Theorem 18, we define the following

probability densities, where $\epsilon_2 \geq \epsilon_1 > 0$:

$$\ell_{\epsilon_1}(v) = \frac{\epsilon_1}{2} e^{-\epsilon_1 |v|}, \tag{5.17}$$

$$\ell_{\epsilon_2|\epsilon_1}(v_2; v_1) = \left[\left(\frac{\epsilon_1}{\epsilon_2}\right)^2 \delta(v_1 - v_2) + \left(1 - \left(\frac{\epsilon_1}{\epsilon_2}\right)^2\right) \right. \tag{5.18}$$

$$\left. \ell_{\epsilon_1}(v_1 - v_2) \right] \frac{\ell_{\epsilon_2}(v_2)}{\ell_{\epsilon_1}(v_1)}, \tag{5.19}$$

$$\ell_{\epsilon_1|\epsilon_2}(v) = \left(\frac{\epsilon_1}{\epsilon_2}\right)^2 \delta(v) + \left(1 - \left(\frac{\epsilon_1}{\epsilon_2}\right)^2\right) \ell_{\epsilon_1}(v), \tag{5.20}$$

where $\delta(\cdot)$ is Dirac's delta function. We refer the reader to earlier work Koufogiannis et al. (2016) on the properties of these distributions. The proof of the following theorem, which proposes a mechanism and proves its privacy guarantees, can be found in the Appendix.

Theorem 18. *Given the sequence of privacy levels $[\epsilon_t]_{t=1}^T$, define the processes $[W_t]_{t=1}^T$ and $[V_t]_{t=1}^T$ such that $V_1 \sim \ell_{\epsilon_1}$ and, for $t \geq 2$,*

- *if $\epsilon_t > |a_t|\, \epsilon_{t+1}$, set*

$$W_t \sim \ell_{\epsilon_{t+1}|\frac{\epsilon_t}{|a_t|}} \quad \textit{and} \quad V_{t+1} = a_t\, V_t - W_t; \tag{5.21}$$

- *if $\epsilon_t \leq |a_t|\, \epsilon_{t+1}$, set*

$$W_t = 0 \quad \textit{and} \quad V_{t+1} | a_t\, V_t \sim \ell_{\epsilon_{t+1}|\frac{\epsilon_t}{|a_t|}}. \tag{5.22}$$

Then,

- *at time t and given the responses $\{\hat{y}_i\}_{i=1}^t$, the current state x_t is ϵ_t–private.*
- *the cost C_T is minimized; i.e. $C_T = \frac{1}{T} \sum_{t=1}^T \frac{2}{\epsilon_t^2}$.*

The proof of this result can be found in the Appendix. Theorem 18 suggests a practical online algorithm. Specifically, at time t, the samples W_t and V_{t+1} depend only on the

current level ϵ_t and the next one ϵ_{t+1}. Additionally, the controller and sensor part of the privacy–preserving mechanism do not need to communicate —the sensor part can infer the noises the controller injects. At each time step, Theorem 18 performs one of the following actions.

- If the current privacy level is tighter than the next one ($\epsilon_t \leq |a_t|\, \epsilon_{t+1}$), then, the sensor performs gradual release of private data according to Koufogiannis et al. (2016), and there is no need to inject any noise to the system.

- If the current privacy level is looser than the next one ($\epsilon_t > |a_t|\, \epsilon_{t+1}$), then, the released information $\hat{y}_t$ can be used to infer the next state x_{t+1} and, thus, violating the privacy guarantees. Theorem 18 enforces privacy by injecting noise and driving the next state of the system x_t away from the predicted one $a_t\, \hat{y}_t$.

At each time, Theorem 18 publishes accurate responses $\hat{y}_t$ of the current state $y_t = x_t$. Specifically, any other proxy with smaller expected squared error $\mathbb{E}(\hat{y}_t - y_t)^2$ would violate the privacy constraints. Nonetheless, the algorithm does not penalize the use of input noise and, therefore, minimizes the quadratic cost C_T which penalizes inaccurate responses. However, the cost C_T does not penalize the noise W_t added to the private data.

Moreover, Theorem 18 is amenable to an infinite horizon setting since, intuitively, the privacy budget is regenerated by the input noise W_t.

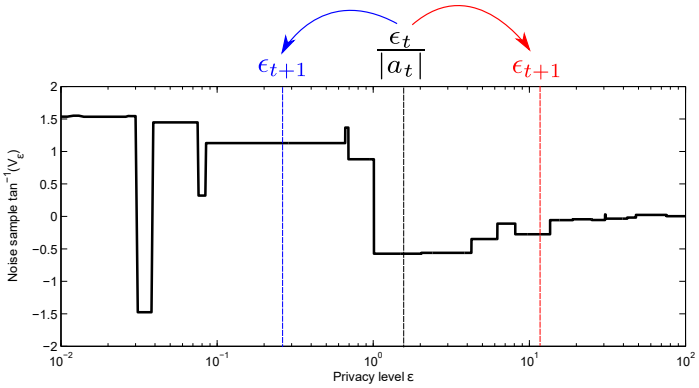


Figure 12: Theorem 18 can be understood in terms of the stochastic process introduced in Koufogiannis et al. (2016). At each time step, we either perform gradual release of private data (denoted by red) and publish a more accurate reponse, or we tighten the privacy by perturbing the private data itself (denoted by blue).

CHAPTER 6: Location–dependent Privacy

6.1. Motivation

Location-based services (LBSs) are daily used by many individuals. In a typical scenario, users retrieve their exact location using a GPS sensor, report it to a provider of an LBS, and receive information regarding this location. For example, a user might request information about nearby places of interest, such as gas stations and restaurants, or subscribe to alert notifications, such as extreme weather and traffic conditions.

From a privacy point of view, reporting the exact GPS location poses a privacy threat to the users and possibly deters them from using LBSs altogether. These privacy issues can be mitigated if users perturb their exact location before using an LBS Andrés et al. (2013). By reporting a noisy GPS location, user's exact position cannot be confidently inferred. On the other hand, the utility users receive from using the LBS does not dramatically deteriorate when a perturbation is applied. Indeed, for example, consider a user on a highway inquiring for nearby gas stations. A perturbation of the user's location by a few miles is unlikely to significantly affect the response by such an LBS. Nonetheless, for a user within an urban environment such a perturbation possibly renders the responses from an LBS useless; within city bounds, a perturbation of the user's location by a few blocks is enough to provide privacy without significantly distorting the response of the LBS. Therefore, the amount of the required, yet acceptable, noise varies and depends on the private location itself.

Providing privacy guarantees for users' locations has been studied in the literature. For example, authors in Shokri et al. (2011) consider *mobile* users and an adversary that, given a training set of traces, attempts to track them. The privacy is then defined by quantifying the effectiveness of the adversary's best inference attack in a game–theoretic approach. Another method was proposed in Hoh et al. (2008) where users aggregate their traces using cloaking techniques to provide privacy guarantees. Within differential privacy, the privacy

of individuals' locations is termed *"geoindistiguishability"* in Andrés et al. (2013), where the authors consider *stationary* users interacting with LBSs and bound the privacy loss due to the quantization of the GPS signal by the discritization grid of the map. In a different line of work, Le Ny et al. (2014) consider the setting where, given individuals' GPS locations, the data curator directly publishes a privacy–preserving traffic map. In this work, we wish to publish a privacy–preserving version of the user's location itself, similarly to Andrés et al. (2013). However, we relax the assumption of a single uniform privacy level everywhere. Instead, in the proposed approach, we allow the privacy level to vary across the map. More generally, allowing the privacy level to depend on the private data is beneficial in settings such as the one considered in Hsu et al. (2014), where a single scalar cannot capture the privacy needs over the whole space of private data.

Specifically, let $(\mathcal{U}, \|\cdot\|)$ be a normed space which includes the set of possible private data and let $q : \mathcal{U} \to \mathcal{Y}$ be a deterministic query of interest, where $\mathcal{Y}$ is the set of possible responses. In our case, we will focus on users reporting their locations to LBSs and, thus, we will mostly focus on Euclidean spaces $\left(\mathbb{R}^2, \|\cdot\|_2\right)$ and identity queries $q(u) = u$. Moreover, we consider a *privacy level map* $\epsilon : \mathcal{U} \to \mathbb{R}_+$ where $\epsilon(u)$ quantifies the need for privacy in a neighborhood of u —smaller values of $\epsilon(u)$ correspond to stronger privacy needs. Then, we wish to design a mechanism Q which outputs a noisy approximation $y = Q(u)$ of the private data u which is *"$\epsilon(u)$-differential private around private data u"*. Note that for constant privacy level maps $\epsilon(u) = \epsilon_0$, our problem reduces to standard ϵ-differential privacy.

There are several practical scenarios where an input-dependent privacy level is meaningful. Specifically, we mention the following two examples:

- *Location-based services:* We consider users interacting with an LBS, as a running example throughout the paper. Whenever the users report their location u with ϵ-privacy, they release an approximation $y = u + V$, where the noise V is proportional to ϵ^{-1}. However, in practice, the desired privacy level ϵ depends on the location u itself. Specifically, as illustrated in Figure 13, densely-populated areas achieve

Figure 13: Within densely populated areas (user A), a small perturbation of the exact but private GPS location provides significant privacy. On the contrary, user B requires a larger perturbation in a sparsely-populated area. The figure is adapted from Statistics Canada.

sufficient privacy by using larger values of privacy level. Conversely, a user can more easily be identified in less-crowded areas unless a smaller value of privacy level ϵ is used. By allowing the privacy level to depend on the user's location itself, we can design a single mechanism that satisfies the privacy needs over all regions.

- *Data-dependent incentives:* From the system designer's perspective, having a fixed privacy level for all possible data inputs might not be possible. We depict this idea by sketching the following scenario. Assume that users' private data capture their wealth $u \in [0, 1]$, e.g., quantile of income distribution. Then, when users report their private data, people with $u \to 0$ may require a tight privacy level to protect their privacy, whereas people with $u \to 1$ might benefit by an increased accuracy of the system and, thus, require larger values for privacy levels. Since people may opt out of using such a system, having a flat privacy level is problematic. Instead, a privacy level *map* $\epsilon : [0, 1] \to \mathbb{R}_+$ captures the needs of all users.

6.2. Problem formulation

In the case of users reporting their location, the private data $u \in \mathbb{R}^2$ is their GPS coordinates and we focus on mechanisms $\mathcal{Q}$ that approximate the private data itself

$$\mathcal{Q}(u) \approx u. \tag{6.1}$$

Nonetheless, the definition of Lipschitz privacy cannot directly capture the problem of input-dependent privacy level, as motivated earlier in this section. Specifically, in Equation (2.5), the privacy level ϵ is a uniform constant in u. Definition 19 alleviates this by using a *privacy level map*.

Definition 19 (Local Lipschtiz Privacy)**.** *Consider a normed space $(\mathcal{U}, \|\cdot\|)$ of private data, a privacy level map $\epsilon : \mathcal{U} \to \mathbb{R}_+$, and a set $\mathcal{Y}$ of possible responses. Then, the mechanism $\mathcal{Q} : \mathcal{U} \to \Delta(\mathcal{Y})$ is $\epsilon(\cdot)$-Lipschitz private if, for any $\mathcal{S} \subseteq \mathcal{Y}$, the function*

$$f_{\mathcal{S}}(u) = \ln \mathbb{P}(\mathcal{Q}(u) \in \mathcal{S}) \tag{6.2}$$

is locally Lipschitz continuous with constant $\epsilon(u)$ for any $u \in \mathcal{U}$.

Locally Lipschitz privacy extends Lipschitz privacy, which implies the standard notion of differential privacy (e.g. Proposition 6 in Koufogiannis et al. (2016)). Specifically, for constant privacy level maps $\epsilon(u) = \epsilon_0$, we retrieve Definition 4. Additionally, Proposition 20 states that, similar to differential privacy, locally Lipschitz privacy is resilient to post-processing; any further processing of the outcome of a locally Lipschitz private mechanism cannot break the privacy guarantees.

Proposition 20. *Let $\mathcal{Q} : \mathcal{U} \to \Delta(\mathcal{Y})$ be a locally Lipschitz mechanism, and $h : \mathcal{Y} \to \mathcal{Z}$ be a (possible randomized) post-processing, where $\mathcal{Y}$ and $\mathcal{Z}$ are two sets of responses. Then, the mechanism $h \circ \mathcal{Q}$ that post-process the outcome of mechanism $\mathcal{Q}$ is ϵ–locally Lipschitz private.*

Proof. The statement follows by re-writing the probability distribution of $h \circ \mathcal{Q}$ in terms of that of $\mathcal{Q}$

$$\mathbb{P}\left((h \circ \mathcal{Q})(u) \in \mathcal{S}\right) = \mathbb{P}\left(\mathcal{Q}(u) \in h^{-1}\left(\mathcal{S}\right)\right) \tag{6.3}$$

and noting that the right–hand side is locally Lipschitz at u with constant $\epsilon(u)$. □

Remark 5. Similarly to the privacy level ϵ in differential privacy, the privacy level map $\epsilon : \mathcal{U} \to \mathbb{R}_+$ in Definition 19 is considered public knowledge and is a designer's choice.

In the light of Definition 19, our problem can be naturally formulated as follows.

Problem 6. *Given a set of private data $(\mathcal{U}, \|\cdot\|)$, a privacy level map $\epsilon : \mathcal{U} \to \mathbb{R}_+$, and a query $q : \mathcal{U} \to \mathcal{Y}$, design an ϵ-locally Lipschitz private mechanism $\mathcal{Q}$ that approximates q.*

6.2.1. Smooth Local Sensitivity

The notion of locally Lipschitz privacy is related to Nissim et al. (2007) which introduced the notion of *smooth local sensitivity* as a mean of building differentially private mechanisms. From a theoretical point of view, we consider our present work as the "dual" of Nissim et al. (2007). Specifically, let $(\mathbb{R}^n, \|\cdot\|)$ be the space of private data and consider a real-valued deterministic query q which mechanism $\mathcal{Q}$ should approximate:

$$q : \mathbb{R}^n \to \mathbb{R}. \tag{6.4}$$

The Laplace mechanism Dwork et al. (2006) allows one to build a private mechanism by adding Laplace-distributed noise as in Proposition 21.

Proposition 21 (Laplace Mechanism)**.** *Consider the Laplace mechanism $\mathcal{Q}$ defined as*

$$\mathcal{Q}(u) = q(u) + V, \text{ with } V \sim Lap\left(\frac{\Delta q^{global}}{\epsilon}\right), \tag{6.5}$$

where $Lap(b)$ is the Laplace distribution with probability density function $f_V(v) = \frac{1}{2b}\, e^{-b\,|v|}$

and Δq^{global} *is the global sensitivity defined as*

$$\Delta q^{global} = \max_{u,u' \colon (u,u') \in \mathcal{A}} |q(u) - q(u')|. \tag{6.6}$$

Then, mechanism $\mathcal{Q}$ *is* ϵ*-differentially private.*

Proposition 21 shows that the ratio

$$\frac{\text{sensitivity}}{\text{privacy level}} = \frac{\Delta q^{\text{global}}}{\epsilon} \tag{6.7}$$

is a key quantity, determines the amount of the injected noise, and is *independent* of the input u. Work in Nissim et al. (2007) considers input-dependent noise by replacing the global sensitivity Δq^{global} by a smooth version of the local sensitivity Δq^{local}, where local sensitivity is defined as

$$\Delta q^{\text{local}}(u) = \max_{u' \colon (u,u') \in \mathcal{A}} |q(u) - q(u')|. \tag{6.8}$$

In our case, the sensitivity is independent of the input; in fact, we will later focus on identity queries which reduces local sensitivity to a constant. Nonetheless, we allow the privacy level ϵ to depend on the private data u and, thus, add input-dependent noise as well.

Although authors in Nissim et al. (2007) introduced smooth local sensitivity as a means to create less noisy but still private mechanisms, we introduce the privacy level map to increase the expressitivity of differential privacy. Moreover, authors in Nissim et al. (2007) use heavy-tailed (polynomially decaying) noise V instead of the exponentially decaying Laplace distribution. In our approach we exploit a link between differential privacy and the eikonal equation in order to numerically design private mechanisms.

6.3. The Eikonal Equation and Results

In the following we focus on Euclidean spaces $(\mathbb{R}^n, \|\cdot\|_2)$ and we focus on building locally Lipschitz private mechanisms that approximate a query $q : \mathbb{R}^n \to \mathcal{Y}$. To this end, we identify the privacy constraint of Definition 19 as an instance of the eikonal equation.

6.3.1. The Eikonal Equation

First, we provide a brief overview of the eikonal equation, a PDE that takes the form of Equation (6.9):

$$\|\nabla u(x)\|_2 = f(x),\ x \in \Omega \text{ and } \ u(x)|_{x\in\partial\Omega} = 0, \tag{6.9}$$

where $\Omega \subseteq \mathbb{R}^n$. The solution $u(x)$ of Equation (6.9) can be thought as the shortest path problem in the continuous domain

$$u(x) = \min_{y\in\Omega} d_f(x, y), \tag{6.10}$$

where d_f is a distance function such that $d_f(x, x + dx) \approx f(x)\,\|dx\|_2$, for small enough dx.

Although the boundary value problem in (6.9) does not always admit strong solutions, literature provides efficient algorithms for computing weak solutions of it. For example, authors in Sethian (1996), Tsitsiklis (1995) introduced the *fast-marching methods* for numerically solving such boundary value problems over discretized grids of N points with complexity $\mathcal{O}(N \log N)$. Following work provided improved algorithms for general meshes Sethian and Vladimirsky (2000) and approaches with accuracy bounds Hassouna and Farag (2007).

By identifying the locally Lipschitz private property in Equation (6.9) as an eikonal equation, we leverage existing, efficient and accurate numerical solvers in order to build locally private mechanisms.

6.3.2. Computing Locally Lipschitz Private Mechanisms

Algorithm 3 proposes a technique to numerically compute locally Lipschitz private mechanisms $\mathcal{Q}$ that approximate a query q.

Algorithm 3 Building a mechanism that satisfies local Lipschitz privacy level map through an eikonal equation solver.

Require: Privacy level map $\epsilon : \mathbb{R}^n \to \mathbb{R}_+$ and query $q : \mathbb{R}^n \to \mathcal{Y}$.

1: **function** PrivacyMapMechanism(Privacy map $\epsilon : \mathcal{U} \to \mathbb{R}_+$, Query $q : \mathcal{U} \to \mathcal{Y}$)

2: **for** each output $y \in \mathcal{Y}$ **do**

3: Compute f_y by solving the eikonal equation problem

$$\|\nabla f_y(u)\| = \epsilon(u) \text{ with } f_y(q^{-1}(y)) = 0. \tag{6.11}$$

4: **end for**

5: Compute $w(y)$ by solving the linear system

$$\sum_{y \in \mathcal{Y}} e^{-f_y(u)}\, w(y) = 1,\ u \in \mathcal{U}. \tag{6.12}$$

6: **if** $w(y) \geq 0$, for all $y \in \mathcal{Y}$ **then**

7: Define mechanism $\mathcal{Q}$ as

$$\mathbb{P}(\mathcal{Q}(u) = y) = w(y)\, e^{-f_y(u)}. \tag{6.13}$$

8: **end if**

9: **end function**

Theorem 22. *Let $(\mathbb{R}^n, \|\cdot\|_2)$ be the space of possible private data and let $q : \mathbb{R}^n \to \mathcal{Y}$ be a query. Then, in Algorithm 3, if*

$$w(y) \geq 0, \quad \forall y \in \mathcal{Y}, \tag{6.14}$$

then, the mechanism $\mathcal{Q}$ is ϵ-locally Lipschitz private.

Proof. The proof is straightforward. The mechanism $\mathcal{Q}$ such that

$$\mathbb{P}(\mathcal{Q}(u) = y) = g(u, y) = w(y)\, e^{-f_y(u)} \tag{6.15}$$

has, by assumption, a proper probability density; $g(u, y) \geq 0$ and $\sum_{y \in \mathcal{Y}} g(u, y) = 1$. More-

over, we compute the following derivative in the weak sense

$$\|\nabla_u \ln \mathbb{P}(\mathcal{Q}(u) = y)\|_2 = \|\nabla_u (\ln\, w(y) - f_y(u))\| = \epsilon(u). \tag{6.16}$$

Therefore, mechanism $\mathcal{Q}$ satisfies Definition 19 and, thus, is ϵ-locally Lipschitz private. □

Algorithm 3 works as follows. For each possible response $y \in \mathcal{Y}$, we solve the following boundary value problem stated in line 3, where the exact boundary condition $f_y(q^{-1}(y)) = 0$ is a design choice. This choice stems from the need that the response y should be close to u, although there is no guarantee that the mode of the resulting distribution $\mathbb{P}(\mathcal{Q}(u) = y)$ is at $y = u$. Next, line 5 of the algorithm computes the weights $w(y)$ such that for each input u, the probability $\mathbb{P}\left(\mathcal{Q}(u) = y\right) = w(y)\, e^{-f_y(u)}$ is a probability distribution. If there exists a positive solution to this linear system, then, the computed mechanism is locally Lipschitz private. As a guideline, for smooth enough privacy maps ($\|\nabla\epsilon(u)\| \ll 1$) with loose privacy at the edge of map ($\epsilon(u)|_{u\in\partial\mathcal{U}} \gg 1$) Algorithm 3 computes well-defined mechanisms.

In practice, Algorithm 3 fails when the privacy level map is not smooth enough although we do not provide sufficient conditions. Nonetheless, for a constant privacy level map, identity queries, and in the limit, we recover the Laplace mechanism.

6.4. Simulations

We demonstrate our technique in the scenario of users reporting their private GPS location to a location-based service (LBS). Specifically, we consider an individual in the greater Philadelphia area that observes her private position $u \in \mathbb{R}^2$, reports a proxy location $y \in \mathbb{R}^2$, and receives a response from the LBS. Due to privacy concerns, the proxy location y is a perturbed version of the exact position u with probability density

$$\mathbb{P}(\mathcal{Q}(u) = y) = g(u, y). \tag{6.17}$$

Under local Lipschitz privacy, we design a privacy level map such that we provide tighter privacy ($\epsilon(u) \to 0$) in sparsely populated areas. To this end, the privacy level map is derived from the population density as

$$\epsilon(u) = 10^{-4}\, d(u) + 0.4, \tag{6.18}$$

where $d(u)$ is the population density at location u. The constant term provides the tightest possible privacy level and the linear term relaxes the privacy level in densely populated areas. The population density map is originated from the *Global Rural-Urban Mapping Project* (GRUMPv1) for International Earth Science Information Network CIESIN et al. (2011) and truly is public knowledge. GRUMPv1 provides an estimate of the population of the whole globe up to a grid size of 30 seconds of arc. —in our case, roughly $0.5\,\text{mi} \times 0.7\,\text{mi}$ rectangles. We focus on an area around Philadelphia of size about $9\,\text{mi} \times 6.2\,\text{mi}$ which is shown in Figure 14. Next, we super-sample this patch to a 200×200 grid, and, for simplicity, we re-parametrize it such that $u \in [0, 100]^2$.

We execute Algorithm 3 for the identity query $q(u) = u$ in Matlab using an eikonal equation solver Dirk-Jan. Algorithm 3 can be run offline and users perturb their private locations by using the stored result. The range of values of the privacy level map is

$$\epsilon(u) \in [0.4, 2.0]. \tag{6.19}$$

Figure 14 shows the probability distributions $\mathbb{P}(\mathcal{Q}(u_i) = y)$ for three different locations. Mechanism $\mathcal{Q}$ adapts to the different values of privacy level for different inputs. Therefore, our approach can provide a single privatizing mechanism without the need to explicitly partition the set of private data.

Finally, we evaluate the performance of the designed mechanism to the following two approaches. To this end, we consider a prior π on the private data u given by the population

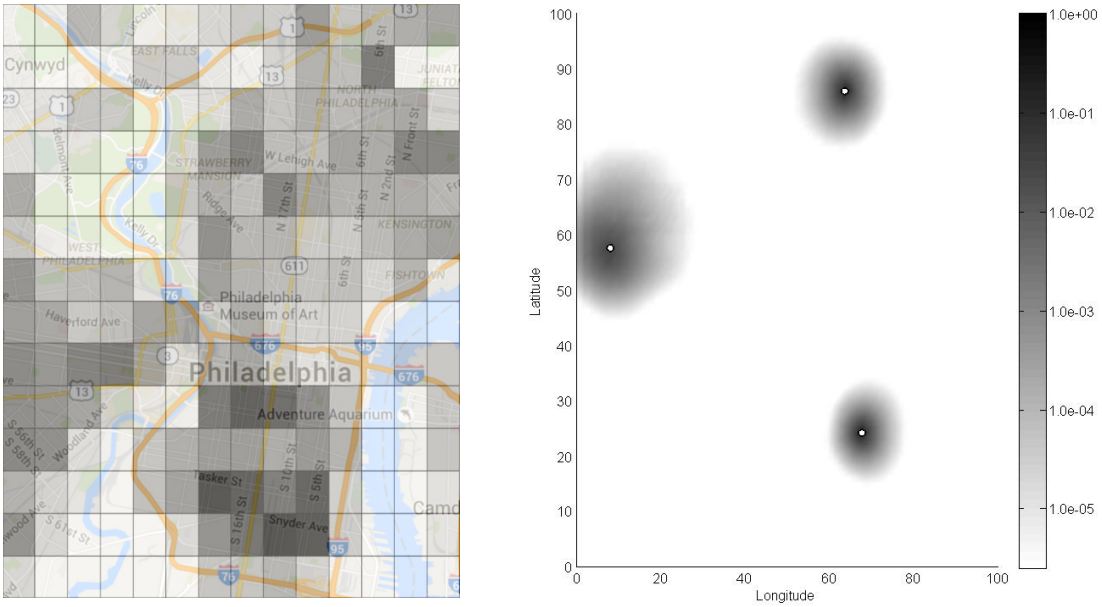


Figure 14: (Left) The population density in Philadelphia's area is shown overlaid with the map is a publicly available knowledge and, thus, has no privacy requirements. In more densely populated areas (darker colored), the privacy level is larger and, thus, less noise is required to mitigate privacy concerns. (Right) The figure shows the probability distribution for three points (denoted with white circles) of high, medium, and low population density as shown in Figure 14. Dense areas have higher values of privacy level and, thus, less amount of noise is required to satisfy the privacy constraint.

	Mean-squared error
$\texttt{mse}_{\text{Laplace}}$	37.5
$\texttt{mse}_{\text{eikonal}}$	5.78
$\texttt{mse}_{\text{optimal}}$	1.37

Table 2: We evaluate the performance of the proposed approach to Laplace-based mechanism and the optimal one.

density itself

$$\pi(u) \propto d(u), \tag{6.20}$$

where d is the population density and we compute the expected mean-squared error of the mechanism $\mathcal{Q}$

$$\texttt{mse}_{\text{eikonal}} = \int_{\mathcal{U}\times\mathcal{U}} \pi(u)\, \mathbb{P}(\mathcal{Q}(u) = y)\, \|u - y\|_2^2 \, du\, dy. \tag{6.21}$$

We compare this to the mean-squared error $\texttt{mse}_{\text{Laplace}}$ of the Laplace mechanism with constant privacy level $\min_{u\in\mathcal{U}} \epsilon(u)$ and to the mean-squared error $\texttt{mse}_{\text{optimal}}$ that is computed by the following optimization problem

$$\begin{aligned} \underset{g:\mathbb{R}^4\to\mathbb{R}_+}{\text{minimize}} \quad & \int_{\mathbb{R}^2\times\mathbb{R}^2} g(u,y)\pi(u)\|y-u\|_2^2 \, d^2u \, d^2u && (6.22)\\ \text{s.t.} \quad & \int_{\mathbb{R}^2} g(u,y)\, d^2y = 1,\ \forall u && (6.23)\\ & \|\nabla_u g(u,y)\| \le \epsilon(u) g(u,y), && (6.24) \end{aligned}$$

where $g(u,y) = \mathbb{P}(\mathcal{Q}_{\text{opt}}(u) = y)$. Due to memory limitations, we solve a coarse (35×35) discritization of Problem 6.22 and report the expected squared-error in Table 2. As expected, a Laplace-based approach injects significant amount of noise which depends on the minimum value of the privacy level map; a single area with tight privacy requirements dramatically affects the performance of the mechanism. Moreover, post-processing the responses of our approach can further improve performance.

CHAPTER 7: Multi–owner Multi–user Privacy

7.1. Introduction

We introduce the problem explored in this chapter with the following example. A collaborative recommender system for merchandise uses the buying history of customers in order to propose an item for future transactions Adomavicius and Tuzhilin (2005). In such a setting, a system that uses the purchase history of Alice to propose new products to her does not harm Alice's privacy. However, whenever Bob's history is used in this process, then, his privacy is compromised. The level of interaction between Alice and Bob is determined by the following two factor: (i) How useful is Bob's data for Alice? In fact, Alice is greatly benefit only if the two agents have similar shopping habits. (ii) How much does Bob trusts Alice? Specifically, if the two agents are close friends, there are little privacy concerns when recommending products to each other.

More generally, the underlying private data curation process is viewed as follows. Multiple agents, called data users, are interested in different aspects of a collection of private data, where the private data is distributed across a set of agents, called data owners—the two sets of agents are not necessarily disjoint. A key observation is that data users may benefit from colluding by either performing their tasks more efficiently or violating owners' privacy.

We now introduce the problem of multi–owner multi–user privacy. Initially, we informally state the general problem and, in the end of this section, we formulate a concrete instance of the problem. We consider two, possibly overlapping, groups of agents: *data owners* and *data users*. Specifically, we consider n data owners with owner $i \in [n] = \{1, \ldots, n\}$ possessing private data $u_i \in \mathcal{U}_i$, where $\mathcal{U}_i$ is the set of possible values for the private data of owner i. Let $\mathbf{u} = [u_i]_{i=1}^n$ denote the set of everyone's private data. We also consider m data users, where each user $j \in [m] = \{1, \ldots, m\}$ is interested in some function $q_j = q_j(\mathbf{u})$ of the private data. Furthermore, we quantify the severity of the privacy concerns that

owner i has against user j with the *privacy level* ϵ_{ij}, where smaller values indicate more severe privacy concerns. Assuming there exists a trusted operator of the system, we wish to design a (randomized) mechanism $\mathcal{M}$ which, given the set of private data $\mathbf{u} \in \times_{i=1}^{n} \mathcal{U}_i$, computes the set of responses $\mathbf{Y} = [Y_i]_{j=1}^{m}$ and securely communicates response Y_j to user j as a proxy of $q_j(\mathbf{u})$. From a utility point of view, we wish each response Y_j to be a good approximation of $q_j(\mathbf{u})$. From a privacy point of view, we wish to guarantee that, given the response Y_j, private data u_i remains ϵ_{ij}–differentially private.

However, data users might decide to collude, share their responses, and violate the privacy needs of a data owner. Therefore, mechanism $\mathcal{M}$ should not incentivized such coalitions. In particular, for any group $\mathcal{J} \subseteq \{1, \dots, m\}$ of data users and any data owner i, there should exist a user $j^* \in \mathcal{J}$ that does not gain any more information about u_i by participating in group $\mathcal{J}$ and, thus, leaves the coalition.

In this work, we focus on the case of real–valued private data u_i, linear queries q_j, and the notion of approximate differential privacy.

7.1.1. SISO to MIMO Privacy

Figure 15 categorizes some of the literature in differential privacy. The vast majority of the literature assumes a single data owner and provides a single response to everyone.

Next, authors of Ebadi et al. (2015) and those of Alaggan et al. (2016) considered n data owners, each with a private data u_i and a privacy level ϵ_i, and propose a mechanism that computes a single output y which is publicly announced. Such a setting, which was also used in Kearns et al. (2016), can be thought as multi–input single-output privacy and, practically, is interpreted as in that, given the response y, each private data u_i remains ϵ_i-private.

Koufogiannis and Pappas (2017a), which was presented in Chapter 3, considered the single input, multiple output case. A single data owner shares a private data u with m users under

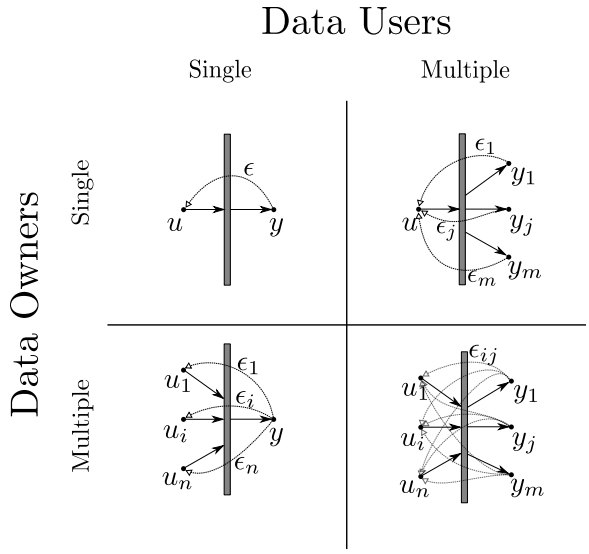


Figure 15: Differential privacy was initially formulated in Dwork et al. (2006) in a SISO way (top-left; T-L); there is a single private data and the operator, shown as a gray bar, computes and publishes a single output. Work in Alaggan et al. (2016), shown in B-L, considered a MISO scenario where different data owners have different privacy requirements and, again, a single output is evaluated and publicly announced. Earlier work Koufogiannis and Pappas (2017a) introduced the case of a single data owner who responds with different privacy levels to different data users (T-R). Here, we consider the multi-input, multi-output case (B-R)

different privacy levels ϵ_j, $j \in [m]$ by responding with y_j to user j. The authors propose a mechanism such that, given y_j, private data u remains ϵ_j-private. More importantly, the proposed mechanism does not incentivize coalitions among the users; i.e. users are not willing to collude and share information in order to damage owner's privacy.

According to such a categorization, present work considers the multi-input, multi-output scenario, where each data owner i has a different privacy level ϵ_{ij} against each data user j. Additionally, each data user is interested in a different aspect of the private data. For example, within a sensor network, based on their location, sensors are interested in mostly local information. As in the SIMO case, it is not enough to guarantee the privacy of owner i against user j and, thus, the MIMO case cannot be decomposed to m MISO systems. Instead, we need to model any possible interactions among the data users that can lead to privacy breaches. Since we assume the existence of a trusted system operator, we focus only on users' interactions that occur after the execution of the differentially private mechanism.

7.1.2. Effects of Coalitions

As briefly mentioned above, the case of designing a private mechanism $\mathcal{M}$, where $\mathcal{M}(\mathbf{u}) = \begin{bmatrix} Y_1 & \cdots & Y_m \end{bmatrix}$, and serve m data users does not decompose into m independent mechanisms $\mathcal{M}_j$, $j \in [m]$, where $\mathcal{M}_j\mathbf{u} = Y_j$. Such a decomposition is not possible due to possible interactions among the data users. In fact, data users may collaborate and exchange information for different reasons. Two models of the possible interaction among data users are the following.

- *Curious coalitions:* Consider a group of data users $\mathcal{J} \subseteq \{1, \ldots, m\}$ and the responses $\mathbf{Y}_\mathcal{J} = [Y_j]_{j\in\mathcal{J}}$ that they receive as a proxy to the quantities of interest $[q_j(\mathbf{u})]_{j\in\mathcal{J}}$. If there is a post-processing of the coalition's knowledge $\mathbf{Y}_\mathcal{J}$ such that *each* colluding user $j \in \mathcal{J}$ extracts a more accurate proxy Y_j' of the quantity of interest $q_j(\mathbf{u})$, then, coalition $\mathcal{J}$ is stable. We call such a group a *curious coalition* because users focus on simply improving the accuracy of their received responses while ignoring any privacy requirements.

- *Adversarial coalitions:* In this case, a subset $\mathcal{J} \subseteq [m]$ of the users collude and share their information $\mathbf{Y}_\mathcal{J}$ in order to infer private data u_i and violate the privacy of a targeted data owner i. In this case, data users are considered adversarial; for example, they might be multiple personae of a single adversary.

Beyond curious and adversarial coalitions, as introduced here, further models exist. For example, in the case that data users also act as data owners, they may or may not care about their own privacy levels whenever they participate in a curious coalition. Specifically, agents participate in a coalition $\mathcal{J}$ only if (i) the accuracy of the received responses is improved and (ii) their privacy level is not compromised, even against other members of the coalition.

In this paper, we will mostly focus on adversarial coalitions. As in the SIMO case Koufogiannis and Pappas (2017a), the main technique against coalitions is introducing correlation

among the responses $\{Y_j\}_{j=1}^m$ that the data users receive. Specifically, the main result of the SIMO case designs a mechanism such that, in each possible coalition, there exists a data user who does not benefit by colluding and, thus, leaves the coalition. Another possible, but not exploited here, technique to de-incentivize coalitions is considering mechanisms that return some side information Z_j to data user j. Then, response Y_j is used by honest users, whereas, side information Z_j is *"used to gratify the curiosity"* of dishonest users.

7.1.3. Linear Queries of Real-Valued Private Data

Here, we assume that each data owner i possesses a scalar private data $u_i \in \mathbb{R}$ and each data user j is interested in a linear form of all the private data

$$q_j(\mathbf{u}) = \sum_{i=1}^{n} a_{ij} u_i. \tag{7.1}$$

Moreover, we assume that each data owner i requires $(\epsilon_{ij}, \delta_{ij})$-differential privacy against data user j. To that end, a trusted system operator receives all private data $\mathbf{u}$, computes the desired quantities $\{q_j(\mathbf{u})\}_{j=1}^m$, adds noise $\mathbf{V}$, and returns the response $Y_j = q_j(\mathbf{u}) + V_j$ to user j:

$$\mathbf{Y} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_m \end{bmatrix} = \begin{bmatrix} q_1(\mathbf{u}) + V_1 \\ \vdots \\ q_m(\mathbf{u}) + V_m \end{bmatrix} = A\,\mathbf{u} + \mathbf{V}, \tag{7.2}$$

where $A = [a_{ij}]$ is the matrix of the coefficients and $\mathbf{V}$ is a privacy-enforcing noise. Now, the problem of MIMO privacy can be formulated.

Problem 7 (MIMO Privacy)**.** *Consider a set of data users $[n]$ and a set of data users $[m]$. For any user $j \in [m]$, any subset of users $\mathcal{J} \subseteq [m]$, and any data owner $i \in [n]$, consider the mechanism in Equation* (7.2)*, and let $\mathcal{M}_{ij}$ be the sub-mechanism that releases Y_j and*

let $\mathcal{M}_{i\mathcal{J}}$ be the sub-mechanism that releases $\{Y_j\}_{j\in\mathcal{J}}$, i.e.

$$\mathcal{M}_{ij}(u_i) = Y_j \quad \textit{and} \quad \mathcal{M}_{i\mathcal{J}}(u_i) = [Y_j]_{j\in\mathcal{J}}. \tag{7.3}$$

Design the noise $\mathcal{V}$ such that,

- *the mechanism $\mathcal{M}_{ij}$ is $(\epsilon_{ij}, \delta_{ij})$-private and*
- *for any group $\mathcal{J}$ and any owner i, there exists a user $j^* \in \mathcal{J}$ such that, if $\mathcal{M}_{ij^*}$ is (ϵ, δ)-private, then, $\mathcal{M}_{i\mathcal{J}}$ is also (ϵ, δ)-private.*

The first constraint of Problem 7 protects each owner's private data against each user, whereas, the second constraint provides privacy against adversarial coalitions. Specifically, for any coalition $\mathcal{J}$ and any target owner i, there exists a data user j^* who does not gain any additional knowledge about owner i (in the sense of privacy level) by participating in the coalition $\mathcal{J}$ and, thus, opts out of it.

7.2. Design via Semidefinite Programming

In this section, we employ the Gaussian mechanism and build a solution to a relaxed version of Problem 7. Specifically, we assume that the system operator adds Gaussian noise with zero mean and covariance matrix $\Sigma \in \mathbb{S}^m$, where $\mathbb{S}^m$ is the set of positive-semidefinite matrices of size $m \times m$, i.e.

$$\mathbf{Y} = A\,\mathbf{u} + \mathbf{V}, \quad \text{where} \quad \mathbf{V} \sim \mathcal{N}(\mathbf{0}_{m\times 1}, \Sigma). \tag{7.4}$$

7.2.1. Analysis of a Coalition

In order to provide an approach to Problem 7, we need to analyze the effect of a coalition. To this end, consider a coalition of data users $\mathcal{J} \subseteq [m]$ and a target data owner $i \in [n]$. For the mechanism defined in Equation (7.4), the following lemma characterizes the privacy level that owner i receives against the group $\mathcal{J}$.

Lemma 23. *For a coalition $\mathcal{J}$ and a target owner i, the mechanism $\mathcal{M}_{i\mathcal{J}}$,*

$$\mathcal{M}_{i\mathcal{J}}(u_i) = [Y_j]_{j\in\mathcal{J}}, \tag{7.5}$$

is (ϵ, δ)-private if

$$\kappa^2(\epsilon, \delta) \geq a_{i\mathcal{J}}^T \, \Sigma_{\mathcal{J}}^{-1} \, a_{i\mathcal{J}}, \tag{7.6}$$

where $a_{i\mathcal{J}} = [a_{ij}]_{j\in\mathcal{J}} \in \mathbb{R}^{|\mathcal{J}|}$ and $\Sigma_J = [\Sigma_{jk}]_{j,k\in\mathcal{J}} \in \mathbb{S}^{|\mathcal{J}|}$.

Sketch of proof. We focus on the following probability as a function of u_i

$$\mathbb{P}\left(\mathcal{M}_{i\mathcal{J}}(u_i) = [y_j]_{j\in\mathcal{J}}\right). \tag{7.7}$$

and we re-write the responses y_i as noisy observations of the private data u_i

$$y_j = \sum_{k\in[n]} a_{kj} \, u_k + V_j \Leftrightarrow \tag{7.8}$$

$$\frac{y_j}{a_{ij}} - \sum_{k\in[n]} \frac{a_{kj}}{a_{ij}} u_k = u_i + \frac{1}{a_{ij}} V_j \Leftrightarrow \tag{7.9}$$

$$z_{ij} = u_i + \frac{1}{a_{ij}} V_j, \quad \forall j \in \mathcal{J} \tag{7.10}$$

where z_{ij} is considered an observation; it does not depend on the noise V_j or the private data u_i. Next, consider the optimal Bayesian linear estimator

$$\hat{u}_i = \sum_{j\in\mathcal{J}} w_j \, z_{ij} = u_i + \sum_{i\in\mathcal{J}} \frac{w_j}{a_{ij}} V_j, \tag{7.11}$$

for appropriate weights w_j with $\sum_{j\in\mathcal{J}} w_j = 1$ and let $\{o_1, \ldots, o_{|\mathcal{J}|-1}\}$ be $|\mathcal{J}| - 1$ orthogonal to $\hat{u}_i$ linear combinations of the observations z_{ij}. Then, the mechanism in Equation (7.7) can be viewed as the mechanism that releases $\hat{u}_i$ as in Equation (7.11) followed by a post-processing which appends to $\hat{u}_i$ the *independent* responses $\{o_1, \ldots, o_{|\mathcal{J}|-1}\}$.

The statement follows from observing that the mechanism in Equation (7.11) is a Gaussian mechanism with variance $\left(a_{i\mathcal{J}}^T \, \Sigma_{\mathcal{J}}^{-1} \, a_{i\mathcal{J}}\right)^{-1}$ and the resilience to post-processing theorem Dwork et al. (2006).

A formal but less intuitive proof follows by massaging Equation (7.7)

$$\mathbb{P}\left(\mathcal{M}_{i\mathcal{J}}(u_i) = [y_j]\right) = \mathbb{P}\left(V_j = a_{ij} \, z_{ij} - a_{ij} \, u_i\right) \tag{7.12}$$

$$\propto e^{-\frac{1}{2}[a_{ij} \, z_{ij} - a_{ij} \, u_i]^T \Sigma_J^{-1} [a_{ij} \, z_{ij} - a_{ij} \, u_i]} \tag{7.13}$$

$$= e^{-\frac{1}{2} C_1 \, u_i^2 + C_2 \, u_i - \frac{1}{2} C_3}, \tag{7.14}$$

with

$$C_1 = [a_{ij}]^T \, \Sigma_{\mathcal{J}}^{-1} \, [a_{ij}], \quad C_2 = [z_{ij}]^T \, \Sigma_{\mathcal{J}}^{-1} \, [a_{ij}], \tag{7.15}$$

$$C_3 = [z_{ij}]^T \, \Sigma_{\mathcal{J}}^{-1} \, [z_{ij}], \tag{7.16}$$

where, for clarity, we have dropped the subscript $j \in \mathcal{J}$ in all stacked vectors, e.g. $[y_j]$. Next, we compare Equation (7.12) to the probability density of a Gaussian mechanism $\mathcal{W}$ that releases $u_i + W$, where $W \sim \mathcal{N}(0, \sigma_w^2)$,

$$\mathbb{P}(u_i + W = w) \propto e^{-\frac{1}{2} u_i^2 \, \sigma_w^{-2} + u_i \, w \, \sigma_w^{-2} - \frac{1}{2} w^2 \sigma_w^{-2}}, \tag{7.17}$$

and we identify the terms

$$\sigma_w^{-2} = [a_{ij}]^T \Sigma_J^{-1} [a_{ij}], \quad w = \frac{[a_{ij}]^T \Sigma_{\mathcal{J}}^{-1} [a_{ij} z_{ij}]}{[a_{ij}]^T \Sigma_{\mathcal{J}}^{-1} [a_{ij}]}. \tag{7.18}$$

Since Equation (7.17) satisfies the (ϵ, δ)-privacy constraint if $\kappa(\epsilon, \delta) \leq \sigma_w^{-1}$, then, Equation (7.7) also satisfies the (ϵ, δ)-privacy constraint and, thus, $\mathcal{M}_{i\mathcal{J}}$ is (ϵ, δ)-private.

□

7.2.2. Design of Covariance Matrix

We now formulate the problem of designing the covariance matrix Σ which provides a solution to Problem 7. Specifically, we provide a solution to Problem 7 by formulating the optimization problem in Theorem 24.

Theorem 24. *Consider n data owners with private data $\mathbf{u} = [u_i]_{i\in[n]} \in \mathbb{R}^n$ and m data users where $\kappa_{ij} = \kappa(\epsilon_{ij}, \delta_{ij})$ is the privacy level of owner i against user j. Then, consider the mechanism $\mathcal{M}$ that securely returns Y_j to user j*

$$\mathcal{M}\mathbf{u} = A\,\mathbf{u} + \mathbf{V} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_m \end{bmatrix}, \tag{7.19}$$

where $\mathbf{V} \sim \mathcal{N}(\mathbf{0}_{m\times 1}, \Sigma)$ and $\Sigma \in \mathbb{S}^m$ satisfies the constraints

$$\Sigma_{jj} \geq \frac{a_{ij}^2}{\kappa_{ij}^2},\ \forall i \in [n],\ j \in [m] \quad \textit{and} \tag{7.20}$$

$$\frac{1}{a_{i\mathcal{J}}^T\,\Sigma_{\mathcal{J}}^{-1}\,a_{i\mathcal{J}}} \geq \min_{j\in\mathcal{J}} \frac{\Sigma_{jj}}{a_{ij}^2},\ \forall \mathcal{J} \subseteq [m],\ i \in [n]. \tag{7.21}$$

Then, mechanism $\mathcal{M}$ is multi-input multi-out private. Specifically, $\mathcal{M}$ satisfies the privacy requirements and does not incentivize adversarial coalitions.

Sketch of proof. The first set of constraints follows from the Gaussian mechanism and requires that owner's i data remains private from user j. The second set of constraints refers to the correlation of the responses that different users receive and is interpreted as follows. For any coalition $\mathcal{J}$ and any targeted owner i, according to Lemma 23, the most privacy-violating inference of the adversarial coalition has variance

$$\frac{1}{a_{i\mathcal{J}}^T\,\Sigma_{\mathcal{J}}^{-1}\,a_{i\mathcal{J}}}. \tag{7.22}$$

We, then, require that there exists a colluding user $j^* \in \mathcal{J}$ that already has inferred user's

i data u_i with smaller variance. □

Theorem 24 provides necessary conditions for a mechanism to be MIMO private. On the downside, Theorem 24 has exponentially many in the number of users constraints. Additionally, the covariance matrix may be over-constrained and, thus, the feasibility problem might be infeasible. Lastly, the second set of constraints is non-convex which makes the design of the covariance matrix challenging.

Nonetheless, the expressitivity of Theorem 24 allows focusing only a subset of potential coalitions. For example, the system designer can choose to focus only on coalitions up to fixed size or ignore coalitions across non-cooperative groups of users. In particular, for agents that act both as data owners and data users, we can ignore coalitions where agent i participates and attempt to attack herself.

Next, in Theorem 25 we propose a convex relaxation which provides privacy of each owner from each user and approximately de-incentivizes users' coalitions.

Theorem 25. *In the setting of Theorem 24, let the covariance matrix Σ be the solution of the following optimization problem*

$$\underset{\Sigma\in\mathbb{S}^m}{\textit{minimize}} \quad \|\textit{diag}(\Sigma)\|_p \tag{7.23}$$

$$s.t. \quad \Sigma_{jj} \geq \max_i \frac{a_{ij}^2}{\kappa_{ij}^2},\ \forall j \tag{7.24}$$

$$\begin{bmatrix} D_{i\mathcal{J}}^{-1} & a_{i\mathcal{J}}^T \\ a_{i\mathcal{J}} & \Sigma_{\mathcal{J}} \end{bmatrix} \succeq 0,\ \forall \mathcal{J}, i, \tag{7.25}$$

where

$$D_{i\mathcal{J}} = \min_{j\in\mathcal{J}} \left[a_{ij}^{-2} \max_{l\in[n]} \left(\frac{a_{lj}}{\kappa_{lj}} \right)^2 \right]. \tag{7.26}$$

Then, $\mathcal{M}$ is approximately MIMO private. Specifically, $\mathcal{M}$ provides κ_{ij} privacy of owner i from user j and approximately de-incentives coalitions.

Proof. Let Σ satisfy the constraints of Theorem 24. Then, for any $\mathcal{J} \subseteq [m]$ and $i \in [n]$

$$\frac{1}{a_{i\mathcal{J}}^T \, \Sigma_{\mathcal{J}}^{-1} \, a_{i\mathcal{J}}} \geq \min_{j \in \mathcal{J}} \frac{\Sigma_{jj}}{a_{ij}^2} \geq \min_{j \in \mathcal{J}} \left[\frac{1}{a_{ij}^2} \max_{l \in [n]} \left(\frac{a_{lj}^2}{\kappa_{lj}^2} \right) \right] = D_{i\mathcal{J}}. \tag{7.27}$$

Since $D_{i\mathcal{J}} > 0$ and using Schur's complement we get

$$\frac{1}{a_{i\mathcal{J}}^T \, \Sigma_{\mathcal{J}}^{-1} \, a_{i\mathcal{J}}} \geq D_{i\mathcal{J}} \;\Leftrightarrow\; \begin{bmatrix} D_{i\mathcal{J}}^{-1} & a_{i\mathcal{J}}^T \\ a_{i\mathcal{J}} & \Sigma_{\mathcal{J}} \end{bmatrix} \succeq 0. \tag{7.28}$$

Regarding accuracy, user's j response has variance Σ_{jj} and, thus, we choose an objective function that minimizes the diagonal elements of Σ. This formulation approximates the design of a MIMO private mechanism using a convex semi-definite program. □

7.3. Simulations: Smooth Local Averaging

For a case study, we consider n agents who act both as data owners and data users. Specifically, we assume that each user i has a scalar private data $u_i \in \mathbb{R}$. For example, data u_i can capture the health status of agent i or a privately computed exposure to risk (e.g. debt-to-equity ratio). Then, each agent wishes to estimate the smooth local average $q_i(\mathbf{u})$ of its neighborhood. For instance, such an average captures the probability of an agent getting infected by other nodes or the cascade exposure to risk.

Specifically, we consider n agents each placed at location $x_i \in [0,1]^2$ uniformly randomly. Then, each agent wishes to compute a smooth local average q_i of the private data $\mathbf{u}$,

$$q_i(\mathbf{u}) = \sum_{j \neq i} \|x_i - x_j\|^{-1} \, u_j. \tag{7.29}$$

In words, agent i weighs input more from nearby agents than from more distant ones. Additionally, we assume that the agents are connected by an undirected graph $G = ([n], E)$, where $E \subseteq [n]^2$ captures the friendships; agents i and j are connected with an edge $(i,j) \in E$ whenever they are friends. Then, let $d : [n]^2 \to \mathbb{R}_+$ be a measure of how far away agents i

and j lie in this graph, captures the trust level between the two agents, and quantifies the privacy level required between i and j as follows

$$\epsilon_{ij} = d^{-1}(i,j), \quad \delta_{ij} = .01\,\epsilon_{ij}, \quad \kappa_{ij} = \kappa(\epsilon_{ij}, \delta_{ij}). \tag{7.30}$$

Here, we choose $d(i,j)$ to be the resistance distance Klein and Randić (1993), Xiao and Gutman (2003). Next, we apply Theorem 25 in order to design a MIMO private mechanism. Specifically, we consider the following SDP, where we only consider coalitions of size up to $m_{\max}$ and worst variance of the responses that agents receive.

$$\underset{\Sigma \in \mathbb{S}^n}{\text{minimize}} \quad \max_i \Sigma_{ii} \tag{7.31}$$

$$\text{s.t.} \quad \Sigma_{ii} \geq \max_j \left(\frac{a_{ij}}{\kappa_{ij}}\right)^2, \; \forall i \in [n]; \tag{7.32}$$

$$\begin{bmatrix} D_{i\mathcal{J}}^{-1} & a_{i\mathcal{J}}^T \\ a_{i\mathcal{J}} & \Sigma_{\mathcal{J}} \end{bmatrix} \succeq 0, \tag{7.33}$$

$$\forall\, \mathcal{J} \subseteq [n] \text{ s.t. } |\mathcal{J}| \leq m_{\max} \text{ and } \forall\, i \in [n] \setminus \mathcal{J}. \tag{7.34}$$

We evaluate our approach by computing the worst-case incentive to form a coalition; given any potential coalition $\mathcal{J}$ and any targeted agent i, we compute how much more information the most-informed agent $j^* \in \mathcal{J}$ can extract about the targeted agent by participating in the coalition. Formally, we define Incentive as

$$\textsc{Incentive} := \max_{\substack{i \in [n], \\ i \notin \mathcal{J} \subseteq [n] \\ \text{s.t. } |\mathcal{J}| \leq m_{\max}-1}} \; \min_{\substack{j \notin \mathcal{J} \\ j \neq i}} \frac{\kappa \text{ of } i \text{ from } \mathcal{J} \cup \{j\}}{\kappa \text{ of } i \text{ from } j}, \tag{7.35}$$

where the expression κ of i from j is the privacy level that protects agent's i private data from the response that agent j receives; as a reminder, larger values correspond to less privacy. Specifically, min chooses the most informed agent j^* and the max chooses the worst-case option over all possible coalitions and possible targets. Figure 16 plots this quantity for different sizes of the network $n \in [2, 16]$ and coalition sizes up to $m_{\max} \in [2, 4]$.

Furthermore, we evaluate the effect of performance of the SDP in the following two ways:

- We compare the variance $\sqrt{\Sigma_{ii}}$ of the response Y_i that agent i receives to that a baseline where each user uses the results in Chapter 3 which completely de–incentivizes coalitions of any size. Specifically, Figure 17 plots the ratio IMPROVEMENT defined as

$$\text{IMPROVEMENT} := \max_{i\in[n]} \frac{\sqrt{\sum_{\substack{j\in[n],\\ j\neq i}} \left(\frac{a_{ij}}{\kappa_{ij}}\right)^2}}{\sqrt{\Sigma_{ii}}}, \tag{7.36}$$

  where the numerator is the variance of the baseline. Importantly, the baseline assumes that each agent i retains all the information $\{Z_{ji}\}_{j\in[n]}$ as side information, whereas, the proposed approach does not require agents to retain such side information.

- We compare the proposed method to the case where we do not protect the private data against coalitions. This figure of merit captures how performance degrades in order to defeat coalitions. Figure 18 compares the variance $\sqrt{\Sigma_{ii}}$ that agent i observes to the variance $\max_i \frac{a_{ij}}{\kappa_{ij}}$ that agent i would ideally observe in the absence of the rest of the users:

$$\text{INEFFICIENCY} := \max_{i\in[n]} \frac{\sqrt{\Sigma_{ii}}}{\max_{\substack{j\in[n],\\ j\neq i}} \frac{a_{ij}}{\kappa_{ij}}}. \tag{7.37}$$

In all cases, we averaged the figures of merit over 20 executions.

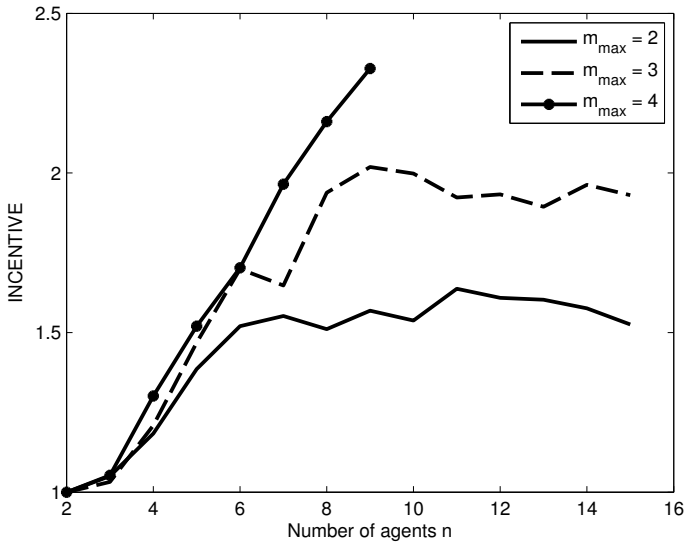


Figure 16: The semidefinite constraints are not binding and, thus, there exists some incentive for agents to form adverserial coalitions. Allowing coalitions of size up to $m_{\max} = 2, 3, 4$, we compute INCENTIVE which captures this gap. Note that INCENTIVE ≥ 1 and larger values result to stronger incentives for agents to collaborate.

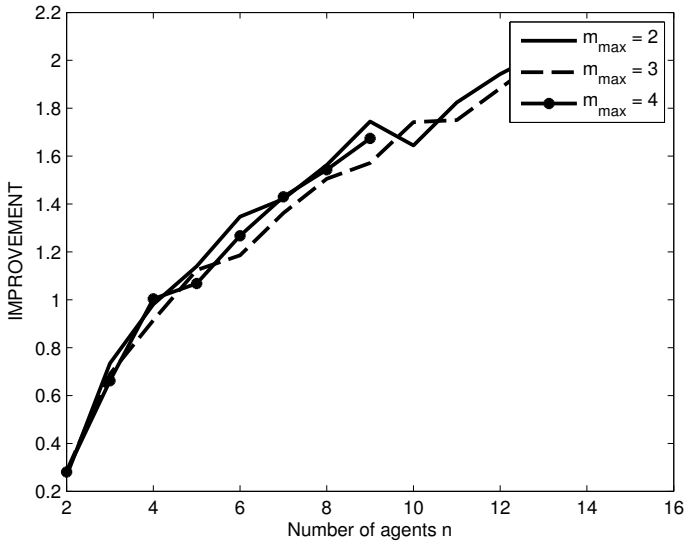


Figure 17: A baseline approach to MIMO privacy utilizes Koufogiannis and Pappas (2017a), where agents independently diffuse their private data. The figure of merit IMPROVEMENT catpures the performance of proposed approach to such a baseline. Althouhg, for very small sizes, the baseline performs better, the proposed approach outperforms the baseline for larger networks.

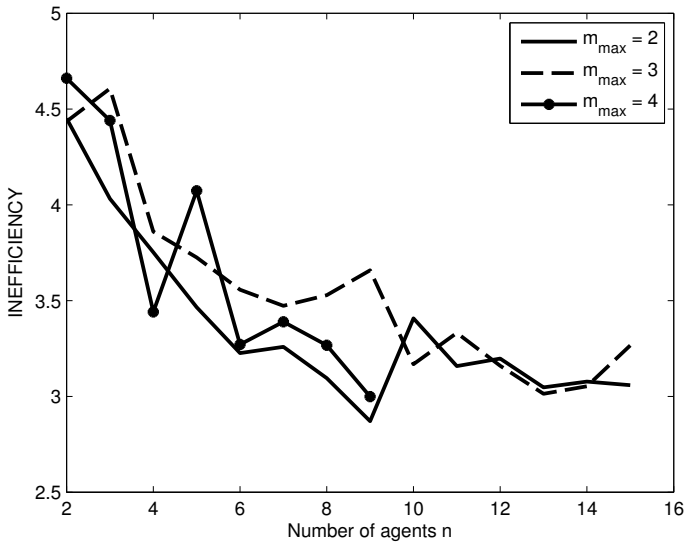


Figure 18: The existence of multiple users force the privacy-enforcing mechanism to inject more noise. We quantify the toll on the accuracy of the responses by plotting INEFFICIENCY which compares the amount of noise added to that of a mechanism that ignores possible coalitions and adds only privacy-preserving noise.

CHAPTER 8: Conclusions and Future Directions

Privacy–preserving techniques are becoming immensely needed in IoT and cloud–based services. Differential privacy provides a formal and robust solution by requiring that the statistics of a published response have controlled dependence on the private data. Despite the wide spectrum of applications within differential privacy, a common assumption states that the private data is fixed, the strength of the privacy needs is predefined, and a single response is published to everyone. The work in this thesis presents problems where these underlying modeling assumptions are relaxed in various ways. Future work can potentially focus on extending differential private mechanisms under such relaxed assumptions. For example, in Chapter 4, gradual release of private data was proven to be possible only for a class of privacy–preserving mechanisms. Extending these results to more sophisticated mechanisms is of interest. Also, Chapter 6 presented a model and a numerical approach to building privacy–preserving mechanisms where the local privacy level depends on the private data itself. Although the so–called privacy level map was assumed given, in practice, a technique for designing this map is needed. Additionally, Chapter 7 focused only on linear queries and a specific model under which coalitions form. Extending this work to a broader class of queries and other models of the data users is another important direction. Furthermore, optimality results such as the ones presented in the appendix are scarce or are stated in an asymptotic sense in the literature. The need for optimality results is dictated by the fact that they are often needed in formulating problems similar to those presented in Chapter 3.

APPENDIX A: Optimality Results

Computing the optimal private mechanism for a fixed privacy level ϵ is considered an open problem for the general case. The Laplace mechanism is a special instance of the exponential mechanism (McSherry and Talwar (2007)) for real spaces $(\mathbb{R}^n, \ell_1)$.

Definition 26 (Laplace Mechanism)**.** *Let $(\mathbb{R}^n, \ell_1)$ be the space of private data. The Laplace mechanism is defined as:*

$$Qu = u + V, \text{ where } V \sim e^{-\epsilon\|V\|_1}. \tag{A.1}$$

The Laplace mechanism can be shown to be ϵ-differentially private. In general, however, the Laplace mechanism is suboptimal in the sense of minimum mean-squared error. For the single-dimensional case, the staircase mechanism Geng and Viswanath (2014) is the optimal ϵ-differentially private mechanism; the mechanism which adds noise V whose distribution is shown in Figure 19. However, the Laplace mechanism is widely used and has several optimality results. Specifically, it is proven to be "universally" optimal —optimally approximating a single linear query, under any prior on private data— and, additionally, it is the optimal ϵ-Lipschitz private mechanism in the sense of both minimum entropy Wang et al. (2014) and minimum mean–squared error Koufogiannis et al. (2015), whereas the staircase mechanism fails to satisfy Lipschitz privacy due to its discontinuous probability density function.

A.0.1. Identity Query under ℓ_1–norm

Theorem 27. *Consider the ϵ-Lipschitz private (in $(\mathbb{R}^n, \ell_1)$) mechanism $Q : \mathbb{R}^n \to \Delta(\mathbb{R}^n)$ of the form $Qu = u + V$, with $V \sim g(V) \in \Delta(\mathbb{R}^n)$. Then, the Laplace mechanism that adds noise with density $l_1^n(v) = \left(\frac{\epsilon}{2}\right)^n e^{-\epsilon\|v\|_1}$ minimizes the mean-squared error. Namely, for any*

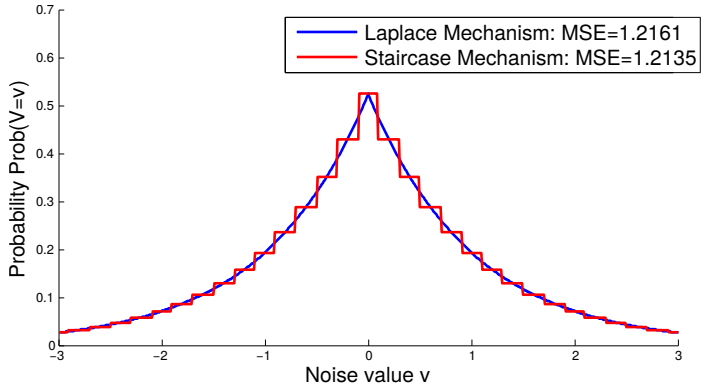


Figure 19: The staircase mechanism is the optimal ϵ-differential private mechanism, whereas the Laplace mechanism is the optimal ϵ-Lipschitz private mechanism. The two distributions are similar and there is only a small performance gap. Therefore, the Laplace distribution is often used in practice.

density g, we have:

$$\mathbb{E}\|Qu - u\|_2^2 = \mathop{\mathbb{E}}_{V\sim g}\|V\|^2 \geq \mathop{\mathbb{E}}_{V\sim l_1^n}\|V\|_2^2 = \frac{2n}{\epsilon^2}. \tag{A.2}$$

For the scalar case ($n = 1$), we give the following proof. A more detailed one can be found in Koufogiannis et al. (2015).

Proof. The optimal mechanism is the solution of the following optimization problem:

$$\begin{aligned} \underset{g\in\Delta(\mathbb{R})}{\text{minimize}} \quad & \mathop{\mathbb{E}}_{V\sim g} V^2 \\ \text{s.t.} \quad & Q \text{ is } \epsilon\text{-Lipschitz private.} \end{aligned} \tag{A.3}$$

The optimization is assumed over the infinite-dimensional space of probability measures over the real line. For a simplified proof, we restrict our attention to probability measures that are continuous and almost everywhere differentiable. This assumption is removed in

the technical proof. The privacy constraint is massaged:

$$\begin{aligned} & Q \text{ is } \epsilon\text{-Lipschitz private } \Rightarrow \\ \left|\frac{d}{du}\ln \mathbb{P}(Qu=y)\right| &\leq \epsilon,\ \forall u, y \Leftrightarrow \\ \left|\frac{d}{du}\mathbb{P}(V=y-u)\right| &\leq \epsilon \mathbb{P}(V=y-u),\ \forall u, y \Leftrightarrow \\ \left|g'(v)\right| &\leq \epsilon g(v),\ \forall v. \end{aligned} \tag{A.4}$$

Specifically, g should be continuous and g' should exist almost everywhere. Problem (A.3) can, then, be restated as a linear program:

$$\begin{aligned} \underset{g:AC(\mathbb{R}\to\mathbb{R}_+)}{\text{minimize}} \quad & \int_{\mathbb{R}} v^2 g(v) dv \\ \text{s.t.} \quad & \int_{\mathbb{R}} g(v) dv = 1, \\ & -\epsilon g(v) \leq g'(v) \leq \epsilon g(v),\ \forall v \in \mathbb{R}, \end{aligned} \tag{A.5}$$

where AC denotes the set of absolutely continuous functions. Problem (A.5) is an infinite-dimensional linear program with uncountably many constraints. We assign the dual variables $\lambda \in \mathbb{R}$ and $\kappa, \mu : \mathbb{R} \to \mathbb{R}_+$ for the two constraints, respectively. The dual of Problem (A.5) is:

$$\begin{aligned} \underset{\lambda\in\mathbb{R},\eta\in\mathcal{C}^1(\mathbb{R}\to\mathbb{R})}{\text{maximize}} \quad & \lambda \\ \text{s.t.} \quad & \eta'(v) + \epsilon|\eta(v)| \leq v^2 - \lambda,\ \forall v \in \mathbb{R}, \\ & \lim_{v\to\infty} \eta(v) \geq 0, \quad \lim_{v\to-\infty} \eta(v) \leq 0. \end{aligned} \tag{A.6}$$

Once both primal Problem (A.5) and dual Problem (A.6) are stated, we construct primal and dual feasible solutions, summon weak duality, and establish optimality. The Laplace distribution $g(v) = \frac{\epsilon}{2}e^{-\epsilon|v|}$ is a primal feasible solution for Problem (A.5) with cost $\frac{2}{\epsilon^2}$. Moreover, we construct a dual feasible solution for Problem (A.6) with cost arbitrarily close to $\lambda^* = \frac{2}{\epsilon^2}$. Specifically, for any $\lambda < \lambda^*$, we are able to construct a dual feasible

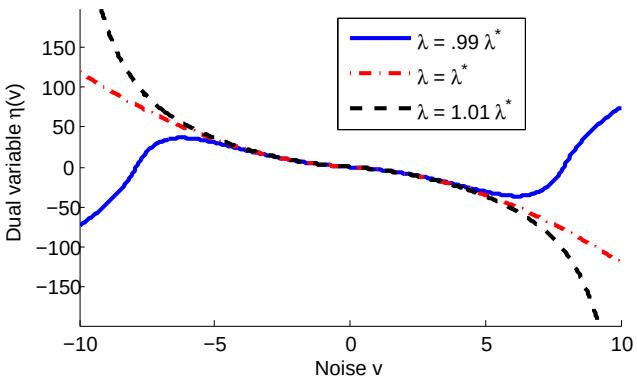


Figure 20: The dual variable $\eta(v)$ is the solution to the intial value problem $\eta'(v)+\epsilon|\eta(v)| = v^2 - \lambda$, $\eta(0) = 0$ for different values of λ. A feasible solution needs to satisfy the boundary constraint $\lim_{v\to\infty} \eta(v) \geq 0$. For $\lambda < \lambda^*$, the solution η is feasible.

solution (λ, η) that satisfies the initial value problem:

$$\eta(0) = 0 \text{ and } \eta'(v) + \epsilon|\eta(v)| = v^2 - \lambda, \ \forall v \in \mathbb{R}\backslash\{0\}. \tag{A.7}$$

Figure 20 plots the unique solution $\eta : \mathbb{R} \to \mathbb{R}$ of the initial value problem (A.7) for different values of λ. For $\lambda < \lambda^*$, the unique solution η of the initial value problem (A.7) is feasible since it satisfies the boundary constraints:

$$\lim_{v\to\infty} \eta(v) \geq 0, \quad \lim_{v\to-\infty} \eta(v) \leq 0. \tag{A.8}$$

On the contrary, the dual variable η is infeasible for $\lambda \geq \lambda^*$. Weak duality establishes the optimality of the Laplace mechanism. Surprisingly, the dual solution $\eta(v) = -\frac{1}{\epsilon^2} v(\epsilon|v| + 2)$ for the optimal value λ^* is infeasible. The infinite dimensionality of the problem leads to an open set of feasible solutions for problem (A.6) and generates this paradox.

□

For higher dimensions, we leverage the result for $n = 1$.

Proof. The optimal mechanism is the solution of the following optimization problem:

$$\begin{aligned} \underset{g:AC(\mathbb{R}^n\to\mathbb{R}_+)}{\text{minimize}} \quad & \int_{\mathbb{R}^n} g(v)v^T v dv \\ \text{s.t.} \quad & \int_{\mathbb{R}^n} g(v)dv = 1, \\ & \|\nabla g(v)\|_\infty \leq \epsilon g(v),\ \forall v \in \mathbb{R}^n. \end{aligned} \tag{A.9}$$

The last constraint is equivalent to

$$-\epsilon g(v) \leq \frac{\partial g}{\partial v_i} \leq \epsilon g(v),\ \forall v \in \mathbb{R}^n,\ \forall i \in \{1,\dots,n\}. \tag{A.10}$$

We consider the dual variables $\lambda \in \mathbb{R}$ and $\kappa_i, \mu_i : \mathbb{R}^n \to \mathbb{R}_+$, set $\eta_i(v) = \mu_i(v) - \kappa_i(v)$, and derive the dual problem:

$$\begin{aligned} \underset{\lambda\in\mathbb{R},\eta_i\in\mathcal{C}^1(\mathbb{R}^n\to\mathbb{R})}{\text{maximize}} \quad & \lambda \\ \text{s.t.} \quad & \sum_{i=1}^n \left\{ \frac{\partial \eta_i}{\partial v_i} + \epsilon|\eta_i(v)| \right\} \leq \sum_{i=1}^n v_i^2 - \lambda, \\ & \lim_{v_i\to\infty} \eta_i(v) \geq 0,\ \lim_{v_i\to-\infty} \eta_i(v) \leq 0,\ \forall i. \end{aligned} \tag{A.11}$$

The solution $g(v) = \left(\frac{\epsilon}{2}\right)^n e^{-\epsilon\|v\|_1}$ is feasible for the primal Problem (A.9) and features cost $\frac{2n}{\epsilon^2}$. A feasible solution for the dual Problem (A.11) is defined as:

$$\eta_i(v) = \eta_{1\text{D}}(v_i), \quad \lambda = n\lambda_{1\text{D}}, \tag{A.12}$$

where $(\lambda_{1\text{D}}, \eta_{1\text{D}})$ is a feasible dual solution for the single-dimensional case given by the initial value problem (A.7). Therefore, the dual Problem (A.11) admits a feasible solution with cost arbitrarily close to $\frac{2n}{\epsilon^2}$. Weak duality establishes the optimality of the Laplace mechanism. □

A.1. Identity Query under ℓ_2–norm

Differential privacy with respect to the ℓ_1-norm captures privacy against the participation of individual users. The ℓ_2-norm is a more suitable for users that contribute high-dimensional data such as GPS and power consumption traces. Once again, a version of the Laplace mechanism is proven to achieve minimum mean-squared-error among all ϵ-Lipschitz private mechanisms that approximate the identity query by adding oblivious noise:

Theorem 28. *Consider the ϵ-Lipschitz private (with respect to the ℓ_2-norm) mechanism $Q : \mathbb{R}^n \to \Delta(\mathbb{R}^n)$ of the form $Qu = u + V$, with $V \sim g \in \Delta(\mathbb{R}^n)$. Then, the Laplace mechanism that adds noise V with density $g = l_2^n(v) \propto e^{-\epsilon\|v\|_2}$ minimizes the mean-squared error:*

$$\mathop{\mathbb{E}}_{V\sim g}\|V\|^2 \geq \mathop{\mathbb{E}}_{V\sim l_2^n}\|V\|_2^2 = \frac{n(n+1)}{\epsilon^2}. \tag{A.13}$$

Proof. Once again, the optimal private mechanism is posed as an optimization problem:

$$\begin{aligned} \underset{g:AC(\mathbb{R}^n\to\mathbb{R}_+)}{\text{minimize}} \quad & \int_{\mathbb{R}^n} g(v)\, v^T v\, d^n v \\ \text{s.t.} \quad & \int_{\mathbb{R}^n} g(v) d^n v = 1, \\ & \nabla g(v)\cdot \hat{a} \leq \epsilon g(v), \text{ for a.e. } v \in \mathbb{R}^n, \\ & \forall \hat{a} \in \mathbb{R}^n, \|\hat{a}\|_2 = 1, \end{aligned} \tag{A.14}$$

where the last constraint is equivalent to the privacy constraint $\|\nabla g(v)\|_2^* \leq \epsilon g(v)$. Consider the dual variables $\lambda \in \mathbb{R}$ and $\kappa : \mathbb{R}^n \times \mathbb{S}^{n-1} \to \mathbb{R}_+$, where $\mathbb{S}^{n-1} = \{\hat{a} \in \mathbb{R}^n : \|\hat{a}\|_2 = 1\}$.

Moreover, set $\eta(v) = \kappa(v) - \mu(v)$, and formulate the dual problem of Problem (A.14):

$$\begin{aligned} \underset{\lambda\in\mathbb{R},\kappa\in\mathbb{R}^n\times\mathbb{S}^{n-1}\to\mathbb{R}_+}{\text{maximize}} \quad & \lambda \\ \text{s.t.} \quad & \nabla\cdot\left(\int_{\mathcal{S}^n}\hat{a}\kappa(v,\hat{a})d\hat{a}\right) \\ & \quad + \epsilon\int_{\mathcal{S}^n}\kappa(v,\hat{a})d\hat{a} \le v^T v - \lambda, \\ & \lim_{\|v\|_2\to\infty}\int_{\mathcal{S}^n}\hat{a}\cdot v\,\kappa(v,\hat{a})d\hat{a} \ge 0. \end{aligned} \tag{A.15}$$

A feasible solution for the primal problem (A.14) is:

$$g(v) = \frac{\epsilon^n\Gamma\left(\frac{n}{2}+1\right)}{\pi^{\frac{n}{2}}\Gamma(n+1)}e^{-\epsilon\|v\|_2}, \tag{A.16}$$

with mean-squared error $\lambda^* = \frac{n(n+1)}{\epsilon^2}$. On the other hand, there exists a dual feasible solution for Problem (A.15) with cost arbitrarily close to λ^*. Consider a dual feasible solution of the form:

$$\begin{aligned} \kappa(v,\hat{a}) = [\eta(\|v\|_2)]^+\,\delta\left(\hat{a}+\frac{v}{\|v\|_2}\right) \\ + [\eta(\|v\|_2)]^-\,\delta\left(\hat{a}-\frac{v}{\|v\|_2}\right), \end{aligned} \tag{A.17}$$

where δ is Dirac's delta function on the unit n-sphere $\mathbb{S}^{n-1}$, $\eta : \mathbb{R}_+ \to \mathbb{R}$ is a suitable function, and $[\cdot]^+$ and $[\cdot]^-$ are the positive and negative parts of a function, respectively. Then, we can reduce the feasible region of Problem (A.15) and rewrite it as

$$\begin{aligned} \underset{\lambda\in\mathbb{R},\eta:\mathbb{R}_+\to\mathbb{R}}{\text{maximize}} \quad & \lambda \\ \text{s.t.} \quad & \eta'(r) + \frac{n-1}{r}\eta(r) + \epsilon|\eta(r)| \le r^2 - \lambda \\ & \lim_{r\to\infty}\eta(r) \ge 0. \end{aligned} \tag{A.18}$$

Similarly to the proof of Theorem 27, a feasible solution (λ,η) of Problem (A.18) of the

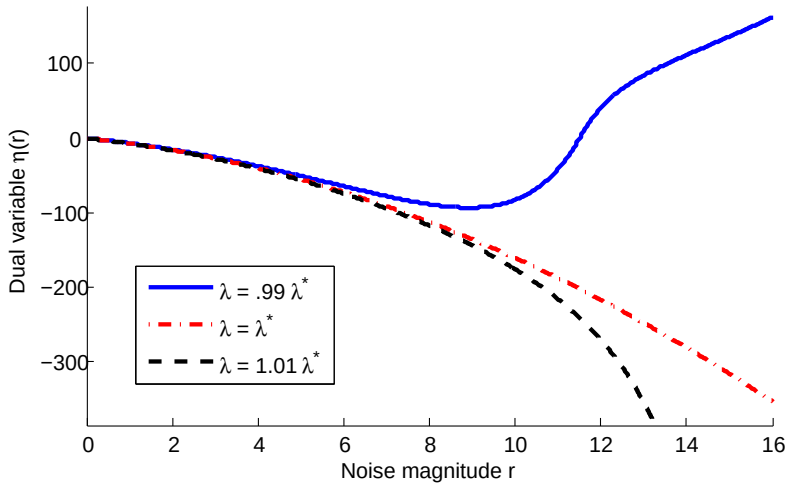


Figure 21: The dual variable $\eta(v)$ is the solution to the intial value problem $\eta'(r)+\frac{n-1}{r}\eta(r)+\epsilon|\eta(r)| = r^2 - \lambda$, $\eta(0) = 0$ for different values of λ. A feasible solution needs to satisfy the boundary constraint $\lim_{v\to\infty}\eta(v) \geq 0$. For $\lambda < \lambda^*$, the solution η is feasible.

following form is constructed:

$$\eta'(r) + \frac{n-1}{r}\eta(r) + \epsilon|\eta(r)| = r^2 - \lambda \text{ and } \eta(0) = 0 \tag{A.19}$$

Figure 21 shows the solution of the initial value problem (A.19) for different values of λ. For $\lambda < \lambda^*$, the solution is feasible and, thus, the optimality of the density (A.16) for the initial value problem (A.14) is established.

Again, for $\lambda = \lambda^*$, the dual solution $\eta(r) = -\frac{r(r\epsilon+n+1)}{\epsilon^2}$ is infeasible as a result of the infinite-dimensional nature of problem (A.19). □

Sample from distribution (A.16) can be efficiently generated. The magnitude $r = \|v\|_2$ of the noise is drawn from the Gamma distribution $r \sim \frac{\epsilon^n}{\Gamma(n)}e^{-\epsilon r}r^{n-1}$ and the direction $\hat{v} = \frac{v}{\|v\|_2}$ is uniformly sampled from the sphere $\mathbb{S}^{n-1}$.

APPENDIX B: Proofs of Chapter 3

Theorem 8 is established in multiple steps. First, we focus on the discrete-domain process $\{V_{\epsilon_i}\}_{i=1}^m$, where $\epsilon_1 \leq \cdots \leq \epsilon_m$ and, in particular, on the case of $m = 2$, with $\epsilon_1 \leq \epsilon_2 < \sqrt{2}\epsilon_1$, where the second inequality is due to technical reasons. Next, we prove the Markov property which allows m discrete privacy levels. Finally, we pass to the limit and derive the continuous-domain process $\{V_\epsilon\}_{\epsilon>0}$ as stated in Theorem 8.

Proof for two privacy levels. We consider the stochastic process V_ϵ supported on two privacy levels $\{\epsilon_1, \epsilon_2\}$, where $\epsilon_1 \leq \epsilon_2 < \sqrt{2}\epsilon_1$. Allowing generalized functions, we assume that the joint distribution of V_{ϵ_1} and V_{ϵ_2} has density:

$$\mathbb{P}(V_{\epsilon_1} = x, V_{\epsilon_2} = y) = l_{\epsilon_1,\epsilon_2}(x, y) \tag{B.1}$$

$$=: g(x, y), \quad x, y \in \mathbb{R}^n \tag{B.2}$$

Density (B.1) should satisfy the following marginal distributions and privacy constraints:

$$\int_{\mathbb{R}^n} g(x, y) d^n y = \epsilon_1^n C_1 e^{-\epsilon_1 \|x\|_2}, \tag{B.3}$$

$$\int_{\mathbb{R}^n} g(x, y) d^n x = \epsilon_2^n C_1 e^{-\epsilon_2 \|y\|_2}, \tag{B.4}$$

$$\|\nabla_x g(x, y) + \nabla_y g(x, y)\|_2 \leq \epsilon_2 \, g(x, y), \tag{B.5}$$

where $C_1 = \frac{\Gamma(\frac{n}{2}+1)}{\pi^{\frac{n}{2}}\Gamma(n+1)}$. The first two constraints express that V_{ϵ_1} and V_{ϵ_2} should be Laplace-distributed with parameters $\frac{1}{\epsilon_1}$ and $\frac{1}{\epsilon_2}$, respectively. The last constraint enforces that the mechanism that releases $(u + V_{\epsilon_1}, u + V_{\epsilon_2})$ must be ϵ_2-private and, thus, the mechanism's log-density needs to be ϵ_2-Lipschitz. We solve for densities g of the form

$$g(x, y) = \epsilon_2^n C_1 \phi(x - y) e^{-\epsilon_2 \|y\|_2}, \tag{B.6}$$

where $\phi : \mathbb{R}^n \to \mathbb{R}$ is a (possibly generalized) function satisfying

$$\begin{aligned} \int_{\mathbb{R}^n} \phi(x-u)\epsilon_2^n e^{-\epsilon_2\|u\|_2} d^n u = \epsilon_1^n e^{-\epsilon_1\|x\|_2}, \\ \int_{\mathbb{R}^n} \phi(u) d^n u = 1. \end{aligned} \tag{B.7}$$

The first equation in (B.7) is a n-dimensional convolution with solution

$$\mathcal{F}\phi(s) = \frac{\mathcal{M}(s;\epsilon_1)}{\mathcal{M}(s;\epsilon_2)}, \tag{B.8}$$

where $\mathcal{M}(s;\epsilon) = \mathcal{F}\left\{\epsilon^n e^{-\epsilon\|x\|_2}\right\}(s)$, and $s \in \mathbb{R}^n$ is the frequency. Solution (B.8) satisfies the second equation in (B.7) since

$$\int_{\mathbb{R}^n} \phi(u) d^n u = \mathcal{F}\phi(s)\big|_{s=0} = \frac{\mathcal{M}(0;\epsilon_1)}{\mathcal{M}(0;\epsilon_2)} = 1. \tag{B.9}$$

Finally, we need to prove that, for ϕ given in (B.8), density g is well-defined, specifically:

$$\phi(z) \geq 0, \quad \forall z \in \mathbb{R}^n. \tag{B.10}$$

This is proven under the assumption that $\epsilon_2 < \sqrt{2}\epsilon_1$; this assumption will eventually be removed. According to Lemma 29, we get:

$$\begin{aligned} \mathcal{F}\phi(s) = \frac{\mathcal{M}(s;\epsilon_1)}{\mathcal{M}(s;\epsilon_2)} &= \left(\frac{\epsilon_1}{\epsilon_2}\right)^{n+1} \left(1 + \frac{\epsilon_2^2 - \epsilon_1^2}{\epsilon_1^2 + \rho^2}\right)^{\frac{n+1}{2}} \\ &= \left(\frac{\epsilon_1}{\epsilon_2}\right)^{n+1} \sum_{k=0}^{\infty} \binom{\frac{n+1}{2}}{k} \left(\frac{\frac{\epsilon_2^2}{\epsilon_1^2} - 1}{1 + \frac{\rho^2}{\epsilon_1^2}}\right)^k, \end{aligned} \tag{B.11}$$

where $\rho = \|s\|_2$. The sum in the right-hand side is an infinite series only when n is even, and, for $\epsilon_2 < \sqrt{2}\epsilon_1$, it converges uniformly in s to the left-hand side. Lemma 30 can be

used to invert the series:

$$\phi(x) = \left(\frac{\epsilon_1}{\epsilon_2}\right)^{n+1} \sum_{k=0}^{\infty} \binom{\frac{n+1}{2}}{k} *^k \left\{ \left(\frac{\epsilon_2^2}{\epsilon_1^2} - 1\right) \epsilon_1^n (2\pi)^{-\frac{n}{2}} (\epsilon_1 r)^{1-\frac{n}{2}} K_{\frac{n}{2}-1}(\epsilon_1 r) \right\}, \tag{B.12}$$

where $r = \|x\|_2$, $K_k(x)$ is the modified Bessel function of the second kind, and $*$ is the n-dimensional convolution. Since $\frac{\epsilon_2^2}{\epsilon_1^2} - 1 \geq 0$ and $K_{\frac{n}{2}-1}(r) \geq 0$, density g is well-defined. $\square$

Next, we prove that the discrete-domain stochastic process $\{V_{\epsilon_i}\}_{i\in\{1,\ldots,m\}}$ is Markov.

Proof of the Markov property. Consider the discrete-domain process $\{V_{\epsilon_i}\}_{i\in\{1,\ldots,m\}}$ supported on m non-decreasing privacy levels $\{\epsilon_1, \ldots, \epsilon_m\}$, and the joint distribution that satisfies the Markov property:

$$\begin{aligned} d\mathbb{P}(V_{\epsilon_i} = v_i,\, \forall i) &= l_{\epsilon_{1:m}}(v_1, \ldots, v_m) \\ &= d\mathbb{P}(V_{\epsilon_1} = v_1) d \prod_{i=2}^{m} \mathbb{P}(V_{\epsilon_i} = v_i | V_{\epsilon_{i-1}} = v_{i-1}) \\ &= l_{\epsilon_1}(v_1) \prod_{i=2}^{m} \frac{l_{\epsilon_{i-1:i}}(v_{i-1}, v_i)}{l_{\epsilon_{i-1}}(v_{i-1})}, \end{aligned} \tag{B.13}$$

where $l_\epsilon(v) \propto e^{-\epsilon\|v\|_2}$ is the n-dimensional Laplace distribution with parameter ϵ^{-1} and $l_{\epsilon_1,\epsilon_2}(v_1, v_2)$ is the density g from the previous proof. Then, the joint distribution $l_{\epsilon_{1:m}}$ satisfies the following properties:

- *Accuracy:* Each coordinate V_{ϵ_i} is optimally-distributed, i.e. Laplace-distributed with

parameter ϵ_i^{-1}:

$$\mathbb{P}(V_{\epsilon_i} = v_k) = \int_{\mathbb{R}^{n(m-1)}} l_{\epsilon_{1:m}}(v_1, \ldots, v_m) dv_{-i} \tag{B.14}$$

$$= l_{\epsilon_i}(v_i), \tag{B.15}$$

where $dv_{-i} = dv_1 \cdots dv_{i-1}\, dv_{i+1} \cdots dv_m$.

- *Privacy:* The mechanism that releases $\{y_i\}_{i=1}^m$, where $y_i = u + V_{\epsilon_i}$ is ϵ_m-private. Indeed, the mechanism can be expressed as:

$$\begin{aligned} \begin{bmatrix} y_1 \\ \vdots \\ y_{m-1} \\ y_m \end{bmatrix} &= \begin{bmatrix} u + V_{\epsilon_1} \\ \vdots \\ u + V_{\epsilon_{m-1}} \\ u + V_{\epsilon_m} \end{bmatrix} \\ &= (u + V_{\epsilon_m}) + \begin{bmatrix} \sum_{i=2}^m V_{\epsilon_{i-1}} - V_{\epsilon_i} \\ \vdots \\ V_{\epsilon_{m-1}} - V_{\epsilon_m} \\ 0 \end{bmatrix}. \end{aligned} \tag{B.16}$$

Density $l_{\epsilon_{i-1},\epsilon_i}$ defined in (B.12) can be re-written in the form

$$\begin{aligned} &l_{\epsilon_{1:m}}(v_1, \ldots, v_m) \\ &\quad = d\mathbb{P}(V_m = v_m) \prod_{i=1}^{m-1} d\mathbb{P}(V_i = v_i | V_{i+1} = v_{i+1}), \end{aligned} \tag{B.17}$$

where

$$d\mathbb{P}(V_i = v_i | V_{i+1} = v_{i+1}) = \frac{\ell_{\epsilon_{i:i+1}}(v_i, v_{i+1})}{\ell_{i+1}(v_{i+1})} \tag{B.18}$$

depends only on the quantity $v_{\epsilon_i} - v_{\epsilon_{i+1}}$. Therefore, V_m is independent of the differences $V_{\epsilon_i} - V_{\epsilon_{i+1}}$. Thus, the mechanism can be viewed as the composition of the ϵ_m-private

mechanism that releases $u + V_{\epsilon_m}$ post-processed by adding independent noise. Since differential privacy is resilient to post-processing Dwork and Roth (2013), the overall mechanism is ϵ_m-private.

□

Finally, we derive the continuous domain process $\{V_\epsilon\}_{\epsilon>0}$ by passing to the limit as the $m \to \infty$, $\epsilon_1 = 0$, and $\epsilon \to \infty$. Specifically, we derive closed-form expressions that lead to efficient algorithms for sampling of the continuous-domain stochastic process.

Proof of the continuous-domain process. In density (B.12), let $\epsilon_1 = \epsilon$ and $\epsilon_2 = (1+\tau)\epsilon$, where $0 < \tau \ll 1$. Then, we prove that we can safely ignore high-order terms:

$$\phi_\epsilon(x) \propto \delta(x) + \mathcal{F}^{-1}\left\{\frac{(n+1)\tau}{1+\frac{\rho^2}{\epsilon^2}}\right\} + O\left(\tau^2\right) \tag{B.19}$$

$$= \delta(x) + \frac{\epsilon^n(n+1)}{(2\pi)^{\frac{n}{2}}}(\epsilon r)^{1-\frac{n}{2}} K_{\frac{n}{2}-1}(\epsilon r)\tau + O\left(\tau^2\right), \tag{B.20}$$

where $r = \|x\|_2$. We discretize a bounded interval $[\underline{\epsilon}, \overline{\epsilon}]$ by considering $K+1$ points $\epsilon^{(i)} = q^i\underline{\epsilon}$, where $q = \left(\frac{\overline{\epsilon}}{\underline{\epsilon}}\right)^{K^{-1}}$, and define the random variable Z as follows:

$$Z := V_{\underline{\epsilon}} - V_{\overline{\epsilon}} = \sum_{i=1}^{K} V_{\epsilon^{(i-1)}} - V_{\epsilon^{(i)}}, \tag{B.21}$$

where the random variables $\{V_{\epsilon^{(i)}}\}_{i=0}^{K}$ form a discrete-domain stochastic process introduced in (B.13). For large K, the step $\tau = q - 1$ becomes arbitrarily small and, thus, we use the first-order approximation in B.19 for each telescoping term $(V_{\epsilon^{(i-1)}} - V_{\epsilon^{(i)}}) \sim \phi_{\epsilon^{(i)}}$. Finally,

the random variable Z is distributed as:

$$Z \sim *_{i=1}^{N} \phi_{\epsilon^{(i)}}(Z) \tag{B.22}$$

$$= *_{i=1}^{N} \Bigg\{ \delta(Z) + \frac{\left(\epsilon^{(i)}\right)^n (n+1)}{(2\pi)^{\frac{n}{2}}} (\epsilon^{(i)} \|Z\|_2)^{1-\frac{n}{2}} \tag{B.23}$$

$$K_{\frac{n}{2}-1}(\epsilon^{(i)}\|Z\|_2)\tau \Bigg\} + O(\tau), \tag{B.24}$$

where we let $\tau \to 0$. This proves that we can approximate the continuous-domain stochastic process by a first-order approximation of the discrete-domain process. □

Equation (B.19) characterizes the stochastic process $\{V_\epsilon\}_{\epsilon>0}$. The atom renders the stochastic process lazy; with high probability, the process is constant over sufficiently small intervals. The linear term governs the statistics of the the jump.

B.1. Proof of Proposition 9

We now provide the proof of Proposition 9 that characterizes the jumps of the stochastic process $\{V_\epsilon\}_{\epsilon>0}$ and, thus, captures the complexity of Algorithm 1.

Proof. Consider the first-order approximation of the backwards conditional distribution ϕ_ϵ derived in (B.19), where $0 < \epsilon$ and $0 < \delta \ll 1$:

$$\mathbb{P}(V_\epsilon = x | V_{(1+\delta)\epsilon} = y) \approx (1 + (n+1)\tau)^{-1} \tag{B.25}$$

$$\left(\delta(x) + \frac{\epsilon^n (n+1)}{(2\pi)^{\frac{n}{2}}} (\epsilon r)^{1-\frac{n}{2}} K_{\frac{n}{2}-1}(\epsilon r)\tau\right) \tag{B.26}$$

Let $a_n(x)$ denote the probability that the process performs n jumps in the interval $[\epsilon, e^x\epsilon]$. Equation (B.25) shows that, for sufficiently small intervals $[\epsilon, (1+\tau)\epsilon]$, the process remains constant with probability $(1 + (n+1)\tau)^{-1}$, therefore, $a_n(x)$ is invariant of ϵ. Under the

discretization introduced earlier, where $\underline{\epsilon} \leftarrow \epsilon$ and $\overline{\epsilon} \leftarrow e^x\epsilon$:

$$a_0(x) = \mathbb{P}(0 \text{ jumps in } [\epsilon, e^x\epsilon]) = e^{-(n+1)x}. \tag{B.27}$$

A limiting argument is used to compute $a_1(x)$:

$$a_1(x) = \lim_{K\to\infty} \sum_{k=1}^{K} \mathbb{P}\left(0 \text{ jumps in } [\epsilon, \epsilon^{(k-1)}]\right) \tag{B.28}$$

$$\mathbb{P}\left(1 \text{ jump in } [\epsilon^{(k-1)}, \epsilon^{(k)}]\right) \mathbb{P}\left(0 \text{ jumps in } [\epsilon^{(k)}, e^x\epsilon]\right) \tag{B.29}$$

$$= (n+1)xe^{-(n+1)x}. \tag{B.30}$$

A similar argument provides a recurrent equation and eventually:

$$a_k(x) = \frac{((n+1)x)^k}{k!} e^{-(n+1)x}. \tag{B.31}$$

Therefore, for a bounded interval interval $[\underline{\epsilon}, \overline{\epsilon}]$, the number n of jumps is characterized by distribution (B.31), which is the Poisson distribution with mean value $(n+1)\ln\left(\frac{\overline{\epsilon}}{\underline{\epsilon}}\right)$. □

B.2. Fourier Transform Pairs

In this section, we derive two Fourier pairs used in the proof of Theorem 8. By convention, the following definition of Fourier transform $f \overset{\mathcal{F}}{\leftrightarrow} F$ is used:

$$\mathcal{F}\{f\}(s) = \int_{\mathbb{R}^n} f(x) e^{-jx\cdot s} d^n x, \tag{B.32}$$

$$\tag{B.33}$$

where $f, F : \mathbb{R}^n \to \mathbb{R}$.

Lemma 29. *The n-dimensional Fourier transform $\mathcal{F}$ of $f : \mathbb{R}^n \to \mathbb{R}$:*

$$f(x) = e^{-\|x\|_2} \tag{B.34}$$

is:

$$\mathcal{F}\{f\}(s) = \frac{\pi^{\frac{n}{2}}\Gamma(n+1)}{\Gamma\left(\frac{n}{2}+1\right)}\left(1+\|s\|_2^2\right)^{-\frac{n+1}{2}}, \tag{B.35}$$

where $s \in \mathbb{R}^n$.

Lemma 30. *The n-dimensional Fourier transform* $\mathcal{F}$ *of* $f : \mathbb{R}^n \to \mathbb{R}$, $f(x) = \|x\|^{1-\frac{n}{2}} K_{\frac{n}{2}-1}(\|x\|)$, *is:*

$$\mathcal{F}\{f\}(s) = \frac{(2\pi)^{\frac{n}{2}}}{1+\rho^2}, \tag{B.36}$$

$$\tag{B.37}$$

where $x \in \mathbb{R}^n$, $\rho = \|s\|_2$, *and* $K_k(z)$ *is the modified Bessel function of the second kind.*

The integrals are formulated using spherical coordinates and, then, symbolically evaluated with Mathematica 10.0. For an non-automated evaluation of the expressions, we refer the reader to MathWorld Weisstein (2015) and references therein, and integral look-up tables Abramowitz and Stegun (1964). We remark that the Bessel function $K_k(z)$ diverges at $z = 0$; for $0 < z \ll 1$, it is $K_k(z) \approx \frac{\Gamma(k)}{2}\left(\frac{2}{x}\right)^k$. Therefore, its Fourier integral converges as the limit of the Laplace transform. This technicality is circumvented here by using look-up tables.

APPENDIX C: Proofs of Chapter 4

We provide a proof of Theorem 15 which allows relaxing the privacy parameters within the framework of approximate differential privacy.

Proof. For a given t, the added noise V_t is distributed according to the Gaussian mechanism for parameters (ϵ_t, δ_t). In order to prove the privacy property, we re-write the released signal as follows:

$$\{Q_\tau u\}_{\tau=-\infty}^{t} = Q_t u + \{V_\tau - V_t\}_{\tau=-\infty}^{t}. \tag{C.1}$$

Mechanism (C.1) can, then, be viewed as the composition of the (ϵ_t, δ_t)-private mechanism with a randomized post-processing. Indeed, the post-processing is independent of the mechanism $Q_t u$ since:

$$B_{\sigma(\epsilon_\tau,\delta_\tau)} - B_{\sigma(\epsilon_t,\delta_t)} \perp B_{\sigma(\epsilon_t,\delta_t)}, \quad \forall \tau \leq t, \tag{C.2}$$

where we used the monotonicity of $\sigma(\epsilon_t, \delta_t)$. □

APPENDIX D: Proofs of Chapter 5

Here, we provide proofs for the two main theorems presented in this work.

Proof of Theorem 16. We will prove the theorem by assuming that the mechanism initially publishes a noisy version $\hat{x}_0$ of the initial state x_0, where

$$\hat{x}_0 = x_0 + E_0, \tag{D.1}$$

where E_0 is artificial noise and we are going to prove the privacy guarantees for such a mechanism that publishes $\hat{x}_0$ and, then, sequentially, $\hat{y}_t$. The post–processing theorem states that the privacy guarantees carry over for the mechanism that does not publish the initial response $\hat{x}_0$.

At time t, given the initial estimator $\hat{x}_0$ and the published responses $[\hat{y}_i]_{i=1}^t$, we denote with $\hat{x}_t$ the least–squares estimator of the current state x_t. For any time t, it suffices to prove that the mechanism that, given x_t as a private data, publishes the least–squares estimator $\hat{x}_t$ is (ϵ_t, δ_t)-differentially private. Indeed, given x_t, the (randomized) function that maps the estimator to the published responses

$$\hat{x}_t \to (\hat{x}_0, \hat{y}_1, \ldots, \hat{y}_t) \tag{D.2}$$

is a post–processing that is independent of the privacy–preserving mechanism that maps the private state to its least–squares estimator

$$x_t \to \hat{x}_t. \tag{D.3}$$

At time $t+1$, for a fixed x_{t+1}, the least–squares estimator $\hat{x}_{t+1}$ is derived as a linear combination of the last estimator $\hat{x}_t = x_t + E_t$ and the last published response $\hat{y}_{t+1} =$

$C\,x_{t+1} + V_{t+1}$. Specifically, letting

$$M_t = A\,\mathbf{\Sigma}_t\,A^T + B\,\mathbf{W}_t\,B^T \text{ and} \tag{D.4}$$

$$N_t = B\,\mathbf{Y}_t - A\,\mathbf{X}_t. \tag{D.5}$$

the least–squares estimator $\hat{x}_{t+1}$ is computed to be

$$\hat{x}_{t+1} = x_{t+1} + K\,V_{t+1} + (I - K\,\mathbf{C})\,(A\,E_t - B\,W_t), \tag{D.6}$$

where $K = (M_t\,C^T + N_t)\,(C\,M_t\,C^T + Z_t + \text{sym}(C\,N_t))^{-1}$ and $\text{sym}(A) = A + A^T$. The covariance of the estimation error $E_{t+1} = \hat{x}_{t+1} - x_{t+1}$ is then

$$\mathbf{\Sigma}_{t+1} = M_t - (M_t\,C^T + N_t)^T \tag{D.7}$$

$$\left(C\,M_t C^T + \text{sym}(C\,N_t) + Z_t\right)^{-1}\,(M_t\,C^T + N_t). \tag{D.8}$$

Next, we relax this equality as follows. The direction of the inequality can be interpreted as the mechanism publishing a more accurate least–squares estimator than the one computed from $\hat{x}_t$ and $\hat{y}_t$. Later, we will demand that this "tighter" estimator meets our privacy requirements.

$$\mathbf{\Sigma}_{t+1} \preceq M_t - (M_t\,C^T + N_t)^T \tag{D.9}$$

$$\left(C\,M_t C^T + \text{sym}(C\,N_t) + Z_t\right)^{-1}\,(M_t\,C^T + N_t) \tag{D.10}$$

We apply Schur complement to retrieve the second inequality in the constraints of (5.10). We complete the proof by invoking the Gaussian mechanism and requiring

$$\mathbf{\Sigma}_{t+1} \succeq \kappa^{-2}(\epsilon_t, \delta_t)\,I. \tag{D.11}$$

□

Proof of Proposition 17. In order to prove feasibility, we need to prove that, for a proper choice of the decision variables, $\mathbf{\Sigma}_t$ has full rank. Then, we can scale any such solution in order to satisfy the privacy constraint. For $\mathbf{Z}_t$ arbitrarily large, i.e. $\mathbf{Z}_t \to \infty$, the second constraint, as stated in the form of Equation D.9 reduces to

$$\mathbf{\Sigma}_t \preceq A\,\mathbf{\Sigma}_t\,A^T + B\,\mathbf{W}_t\,B^T. \tag{D.12}$$

It suffices to prove that the right hand side of the inequality is full rank. Indeed, let $v \in \mathbb{R}^n$ be such that $v^T\,\left(A\,\mathbf{\Sigma}_t\,A^T + B\,\mathbf{W}_t\,B^T\right)\,v = 0$. Then, $v^T\,A\mathbf{\Sigma}_t^{\frac{1}{2}} = 0$ and $v^T\,B\,\mathbf{W}_t^{\frac{1}{2}} = 0$ and, thus, $v^T\,A = 0$ and $v^T\,B = 0$. Since $[A; B]$ has rank n, this implies that $v = 0$ and this completes the proof. □

Proof of Theorem 18. For simplicity, we assume that $a_t \neq 0$. First, we observe that $\mathbb{E}(\hat{y}_t - y_t)^2 = \mathbb{E}\,V_t^2 \geq \frac{2}{\epsilon_t^2}$ due to the optimality of the Laplace mechanism Wang et al. (2014), Koufogiannis et al. (2015). On the other hand, we use induction on t and prove that $V_t \sim \ell_{\epsilon_t}$. For $t = 1$, it holds that $V_1 \sim \ell_{\epsilon_1}$. For $t + 1$, we consider two cases.

- If $\epsilon_t > |a_t|\,\epsilon_{t+1}$, since $V_t \sim \ell_{\epsilon_t}$ and $W_t \sim \ell_{\epsilon_{t+1}|\frac{\epsilon_t}{|a_t|}}$ and are independent, it follows that $V_{t+1} = a_t\,V_t - W_t \sim \ell_{\epsilon_{t+1}}$.

- If $\epsilon_t \leq |a_t|\,\epsilon_{t+1}$, then, by integrating out V_t we get that $V_{t+1} \sim \ell_{\epsilon_{t+1}}$.

Therefore, the minimum cost is achieved and this proves the second part of Theorem 18.

Next, we prove the privacy guarantees using induction on t. We abuse notation by using the symbol $\mathbb{P}$ for probability densities and re-use the same symbol for the random variable and its value Specifically, we prove that, at time t and given the current state x_t, the likelihood probability of the past responses is of the form

$$\mathbb{P}(\hat{y}_1, \ldots, \hat{y}_t) = \ell_{\epsilon_t}(\hat{y}_t - x_t)\,h(\hat{y}_1, \ldots, \hat{y}_t), \tag{D.13}$$

for some function h. Note that the density in Equation (D.13) does not depend on past states $\{x_i\}_{i<t}$. For $t = 1$ and given x_1, it holds

$$\mathbb{P}(\hat{y}_1 = z_1) = \mathbb{P}(V_1 = z_1 - x_1) = \ell_{\epsilon_1}(z_1 - x_1). \tag{D.14}$$

For $t + 1$, we consider two cases.

- If $\epsilon_t > |a_t|\,\epsilon_{t+1}$, we condition on the W_t and from the induction hypothesis we get, given x_{t+1}

$$\begin{aligned} &\mathbb{P}(\hat{y}_1, \ldots, \hat{y}_t | W_t = w) && \text{(D.15)} \\ &\quad = \ell_{\epsilon_t}\left(\hat{y}_t - \frac{x_{t+1} - w}{a_t}\right) h(\hat{y}_1, \ldots, \hat{y}_t). && \text{(D.16)} \end{aligned}$$

  Since $\hat{y}_{t+1} = a_t\,\hat{y}_t$, we compute

$$\begin{aligned} &\mathbb{P}(\hat{y}_1, \ldots, \hat{y}_{t+1}) && \text{(D.17)} \\ &= \int_{\mathbb{R}} \mathbb{P}(\hat{y}_1, \ldots, \hat{y}_t | W_t = w)\,\mathbb{P}(W_t = w)\,dw && \text{(D.18)} \\ &= \ell_{\epsilon_{t+1}}(\hat{y}_{t+1} - x_{t+1})\,h_1(\hat{y}_1, \ldots, \hat{y}_{t+1}), && \text{(D.19)} \end{aligned}$$

  for a function h_2.

- If $\epsilon_t \leq |a_t|\,\epsilon_{t+1}$, given x_{t+1}

$$\mathbb{P}(\hat{y}_1, \ldots, \hat{y}_t) = \ell_{\epsilon_t}\left(\hat{y}_t - \frac{x_{t+1}}{a_t}\right) h(\hat{y}_1, \ldots, \hat{y}_t). \tag{D.20}$$

Then, given x_{t+1}

$$\mathbb{P}(\hat{y}_1, \ldots, \hat{y}_{t+1}) \tag{D.21}$$

$$= \mathbb{P}(\hat{y}_1, \ldots, \hat{y}_t)\, \mathbb{P}(\hat{y}_{t+1} | \hat{y}_t) \tag{D.22}$$

$$= \mathbb{P}(\hat{y}_1, \ldots, \hat{y}_t) \tag{D.23}$$

$$\mathbb{P}\left(V_{t+1} = \hat{y}_{t+1} - x_{t+1} | V_t = \hat{y}_t - \frac{x_{t+1}}{a_t}\right) \tag{D.24}$$

$$= \ell_{\epsilon_{t+1}}(\hat{y}_{t+1} - x_{t+1})\, h_2(\hat{y}_1, \ldots, \hat{y}_{t+1}), \tag{D.25}$$

for a function h_2.

We finish the proof by noting that the log–likelihood function of the responses is ϵ_t-Lipschitz in x_t

$$\left| \frac{d}{dx_t} \ln \mathbb{P}(\hat{y}_1, \ldots, \hat{y}_t) \right| = \epsilon_t. \tag{D.26}$$

□

BIBLIOGRAPHY

M. Abramowitz and I. Stegun. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*. Courier Corporation, 1964.

A. Acquisti and R. Gross. Predicting social security numbers from public data. *Proceedings of the National academy of sciences*, 106(27):10975–10980, 2009.

G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, 2005.

M. Alaggan, S. Gambs, and A.-M. Kermarrec. Heterogeneous differential privacy. *Journal of Privacy and Confidentiality*, 7(2):127–158, 2016.

M. Andrés, N. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2013.

L. Atzori, A. Iera, and G. Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

D. Babić, D. Klein, I. Lukovits, S. Nikolić, and N. Trinajstić. Resistance-distance matrix: A computational algorithm and its application. *International Journal of Quantum Chemistry*, 90(1):166–176, 2002.

J. Bennett, S. Lanning, et al. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York, NY, USA, 2007.

K. Chatzikokolakis, M. Andrés, N. Bordenabe, and C. Palamidessi. Broadening the scope of differential privacy using metrics. In *Privacy Enhancing Technologies*, pages 82–102. Springer, 2013.

V. Costan and S. Devadas. Intel sgx explained. *IACR Cryptology ePrint Archive*, 2016:86, 2016.

K. Dirk-Jan. Accurate fast marching. `http://www.mathworks.com/matlabcentral/fileexchange/24531-accurate-fast-marching`; Accessed on 3/2/2016.

C. Dwork and G. Pappas. Privacy in information-rich intelligent infrastructure. *Computer networks*, 2017.

C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Theoretical Computer Science*, 2013.

C. Dwork and G. N. Rothblum. Concentrated differential privacy. *arXiv preprint arXiv:1603.01887*, 2016.

C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284. Springer, 2006.

H. Ebadi, D. Sands, and G. Schneider. Differential privacy: Now it's getting personal. In *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 2015.

Ú. Erlingsson, V. Pihur, and A. Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067. ACM, 2014.

G. Fanti, P. Kairouz, S. Oh, K. Ramchandran, and P. Viswanath. Hiding the rumor source. *IEEE Transactions on Information Theory*, 2017.

C. for International Earth Science Information Network CIESIN, I. F. P. R. I. IFPRI, and C. I. de Agricultura Tropical CIAT. Global rural-urban mapping project, version 1 (grumpv1): Urban extents grid (africa), 2011. `http://sedac.ciesin.columbia.edu/data/set/grump-v1-urban-extents/maps`; Accessed on 3/2/2016.

F. D. Garcia and B. Jacobs. Privacy-friendly energy-metering via homomorphic encryption. In *International Workshop on Security and Trust Management*, pages 226–238. Springer, 2010.

Q. Geng and P. Viswanath. The optimal mechanism in differential privacy. pages 2371–2375, 2014.

A. Ghosh and A. Roth. Selling privacy at auction. *Games and Economic Behavior*, 91: 334–346, 2015.

A. Ghosh, T. Roughgarden, and M. Sundararajan. Universally utility-maximizing privacy mechanisms. *SIAM Journal on Computing*, 41(6):1673–1693, 2012.

M. Hale and M. Egerstedty. Differentially private cloud-based multi-agent optimization with constraints. In *American Control Conference (ACC)*, 2015.

S. Han, U. Topcu, and G. Pappas. Differentially private convex optimization with piecewise affine objectives. In *IEEE Conference on Decision and Control*, 2014.

S. Han, U. Topcu, and G. J. Pappas. Event-based information-theoretic privacy: A case study of smart meters. In *American Control Conference (ACC), 2016*, pages 2074–2079. American Automatic Control Council (AACC), 2016.

M. Hardt and K. Talwar. On the geometry of differential privacy. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, pages 705–714. ACM, 2010.

S. Hassouna and A. Farag. Multistencils fast marching methods: A highly accurate solution to the eikonal equation on cartesian domains. *Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1563–1574, 2007.

B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Preserving privacy in gps traces via uncertainty-aware path cloaking. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 161–171. ACM, 2007.

B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J. Herrera, A. Bayen, M. Annavaram, and Q. Jacobson. Virtual trip lines for distributed privacy-preserving traffic monitoring. In *Proceedings of the 6th international conference on Mobile systems, applications, and services*, pages 15–28, 2008.

J. Hsu, M. Gaboardi, A. Haeberlen, S. Khanna, A. Narayan, B. C. Pierce, and A. Roth. Differential privacy: An economic method for choosing epsilon. In *Computer Security Foundations Symposium (CSF), 2014 IEEE 27th*, pages 398–410. IEEE, 2014.

J. Hsu, Z. Huang, A. Roth, and Z. S. Wu. Jointly private convex programming. pages 580–599, 2016.

Z. Huang, S. Mitra, and G. Dullerud. Differentially private iterative synchronous consensus. In *Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society*, pages 81–90. ACM, 2012.

Z. Huang, S. Mitra, and N. Vaidya. Differentially private distributed optimization. In *Proceedings of the 2015 International Conference on Distributed Computing and Networking*, page 4. ACM, 2015.

P. Kairouz, S. Oh, and P. Viswanath. Extremal mechanisms for local differential privacy. In *Advances in neural information processing systems*, pages 2879–2887, 2014.

P. Kairouz, S. Oh, and P. Viswanath. The composition theorem for differential privacy. *IEEE Transactions on Information Theory*, 2017.

V. Katewa, A. Chakrabortty, and V. Gupta. Protecting privacy of topology in consensus networks. In *American Control Conference (ACC)*, 2015.

M. Kearns, A. Roth, Z. S. Wu, and G. Yaroslavtsev. Private algorithms for the protected in social network search. *Proceedings of the National Academy of Sciences*, 113(4):913–918, 2016.

D. J. Klein and M. Randić. Resistance distance. *Journal of mathematical chemistry*, 12(1): 81–95, 1993.

F. Koufogiannis and G. J. Pappas. Location-dependent privacy. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pages 7586–7591. IEEE, 2016a.

F. Koufogiannis and G. J. Pappas. Multi-owner multi-user privacy. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pages 1787–1793. IEEE, 2016b.

F. Koufogiannis and G. J. Pappas. Diffusing private data over networks. *IEEE Transactions on Control of Network Systems*, 2017a.

F. Koufogiannis and G. J. Pappas. Differential privacy for dynamical sensitive data. In *Submitted*, 2017b.

F. Koufogiannis, S. Han, and G. Pappas. Computation of privacy-preserving prices in smart grids. In *IEEE Conference on Decision and Control*, 2014.

F. Koufogiannis, S. Han, and G. Pappas. Optimality of the laplace mechanism in differential pricacy. *arXiv preprint arXiv:1504.00065*, 2015.

F. Koufogiannis, S. Han, and G. J. Pappas. Gradual release of sensitive data under differential privacy. *Journal of Privacy and Confidentiality*, 7(2):23–52, 2016.

J. Le Ny and G. Pappas. Differentially private filtering. *IEEE Transactions on Automatic Control*, 2014.

J. Le Ny, A. Touati, and G. Pappas. Real-time privacy-preserving model-based estimation of traffic flows. In *ICCPS'14: ACM/IEEE 5th International Conference on Cyber-Physical Systems*, 2014.

J. Leskovec and J. Mcauley. Learning to discover social circles in ego networks. In *Advances in Neural Information Processing Systems*, 2012.

D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

K. Ligett, S. Neel, A. Roth, B. Waggoner, and Z. S. Wu. Accuracy first: Selecting a differential privacy level for accuracy-constrained erm. *arXiv preprint arXiv:1705.10829*, 2017.

F. McSherry and K. Talwar. Mechanism design via differential privacy. In *IEEE Symposium on Foundations of Computer Science*, 2007.

I. Mironov. Renyi differential privacy. *arXiv preprint arXiv:1702.07476*, 2017.

Y. Mo and R. M. Murray. Privacy preserving average consensus. *IEEE Transactions on Automatic Control*, 62(2):753–765, 2017.

A. Narayanan and V. Shmatikov. How to break anonymity of the netflix prize dataset. *arXiv preprint cs/0610105*, 2006.

K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 75–84. ACM, 2007.

P. Ohm. Broken promises of privacy: Responding to the surprising failure of anonymization. 2009.

P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 223–238. Springer, 1999.

S. Pequito, S. Kar, S. Sundaram, and A. P. Aguiar. Design of communication networks for distributed computation with privacy guarantees. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 1370–1376. IEEE, 2014.

A. Rial and G. Danezis. Privacy-preserving smart metering. In *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, 2011.

L. Sankar, R. Rajagopalan, S. Mohajer, and V. Poor. Smart meter privacy: A theoretical framework. In *IEEE Transactions on Smart Grid*, 2013a.

L. Sankar, S. R. Rajagopalan, and H. V. Poor. Utility-privacy tradeoffs in databases: An information-theoretic approach. *IEEE Transactions on Information Forensics and Security*, 8(6):838–852, 2013b.

A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In *International Workshop on Privacy Enhancing Technologies*, pages 41–53. Springer, 2002.

J. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

J. Sethian and A. Vladimirsky. Fast methods for the eikonal and related hamilton–jacobi equations on unstructured meshes. *Proceedings of the National Academy of Sciences*, 97 (11):5699–5703, 2000.

R. Shokri, G. Theodorakopoulos, J. Le Boudec, and J. Hubaux. Quantifying location privacy. In *Symposium on Security and Privacy*, pages 247–262. IEEE, 2011.

Y. Shoukry, K. Gatsis, A. Alanwar, G. J. Pappas, S. A. Seshia, M. Srivastava, and P. Tabuada. Privacy-aware quadratic optimization using partially homomorphic encryption. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pages 5053–5058. IEEE, 2016.

A. Speranzon and S. D. Bopardikar. An algebraic topological perspective to privacy. In *American Control Conference (ACC), 2016*, pages 2086–2091. IEEE, 2016.

L. Sweeney. Weaving technology and policy together to maintain confidentiality. *The Journal of Law, Medicine & Ethics*, 25(2-3):98–110, 1997.

L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

T. Tanaka, K. Kim, P. Parrilo, and S. Mitter. Semidefinite programming approach to gaussian sequential rate-distortion trade-offs. *arXiv preprint arXiv:1411.7632*, 2014.

T. Tanaka, M. Skoglund, H. Sandberg, and K. H. Johansson. Directed information as privacy measure in cloud-based control. *arXiv preprint arXiv:1705.02802*, 2017.

J. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *Transactions on Automatic Control*, 40(9):1528–1538, 1995.

W. Wang, L. Ying, and J. Zhang. On the relation between identifiability, differential privacy, and mutual-information privacy. *IEEE Transactions on Information Theory*, 62(9):5018–5029, 2016.

Y. Wang, Z. Huang, S. Mitra, and G. Dullerud. Entropy-minimizing mechanism for differential privacy of discrete-time linear feedback systems. In *IEEE Conference on Decision and Control*, 2014.

E. W. Weisstein. Hypergeometric function. from mathworld–a wolfram web resource, 2015. URL `\url{http://mathworld.wolfram.com/HypergeometricFunction.html}`. [Online; accessed 3-June-2015].

Wikipedia. Unix time — wikipedia, the free encyclopedia, 2015. URL `\url{http://en.wikipedia.org/w/index.php?title=Unix_time&oldid=662052467}`. [Online; accessed 13-May-2015].

W. Xiao and I. Gutman. Resistance distance and laplacian spectrum. *Theoretical Chemistry Accounts*, 110(4):284–289, 2003.