

DIGITAL ACCESS TO SCHOLARSHIP AT HARVARD
DASH.HARVARD.EDU

HARVARD LIBRARY


Concurrent Composition of Interactive Mechanisms with Adaptive Privacy-loss Parameters

Citation

Xu, Vicki. 2024. Concurrent Composition of Interactive Mechanisms with Adaptive Privacy-loss Parameters. Bachelors Thesis, Harvard University Engineering and Applied Sciences.

Link

https://dash.harvard.edu/handle/1/42719171

Terms of use

This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Other Posted Material (LAA), as set forth at

https://harvardwiki.atlassian.net/wiki/external/NGY5NDE4ZjgzNTc5NDQzMGIzZWZhMGFlOWI2M2EwYTg

Accessibility

https://accessibility.huit.harvard.edu/digital-accessibility-policy

Share Your Story

The Harvard community has made this article openly available.
Please share how this access benefits you. Submit a story

Concurrent Composition of Interactive Mechanisms with Adaptive Privacy-loss Parameters

A DISSERTATION PRESENTED
BY
VICKI XU
TO
THE DEPARTMENT OF COMPUTER SCIENCE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
BACHELOR OF ARTS
IN THE SUBJECT OF
COMPUTER SCIENCE AND MATHEMATICS

HARVARD UNIVERSITY
CAMBRIDGE, MASSACHUSETTS

NOVEMBER 2023

©2023 – Vicki Xu
all rights reserved.

Thesis advisor: Professor Salil Vadhan Vicki Xu

Concurrent Composition of Interactive Mechanisms with Adaptive Privacy-loss Parameters

Abstract

Over recent decades, the predominance of data analysis algorithms has made privacy an increasingly greater concern. Differential privacy is one framework for providing privacy guarantees for analysis of sensitive data. A persistent research direction in differential privacy is providing theoretical support for all the variety of ways a data analyst can interact with a dataset, so that practical implementations can have provable privacy guarantees.

This thesis is concerned with the setting in which an analyst interacting with a set of differentially-private mechanisms is interested in both adaptively interleaving queries between mechanisms and also creating new ones. Previous work has provided provable guarantees for the sequential composition of non-interactive mechanisms with adaptive privacy-loss parameters, and the concurrent composition of interactive mechanisms with pre-fixed privacy-loss parameters, but no work has addressed the setting in which both the interaction and the privacy-loss parameters can both be chosen adaptively. Hence, we study the concurrent composition of interactive differentially-private mechanisms with adaptively chosen privacy-loss parameters. We provide formulations of privacy filters and odometers, specialized interactive mechanisms that allow for concurrent composition and adaptive composition, and also provide support for privacy-loss tracking. We prove that every valid privacy filter and odometer for non-interactive mechanisms extends to the concurrent composition of interactive mechanisms if privacy loss is measured using (ε, δ)-DP, f-DP, or Rényi DP of fixed order.

Our results offer strong theoretical foundations for enabling full adaptivity in composing differentially private interactive mechanisms, showing that concurrency does not affect the privacy guarantees. This thesis allows simplifications for existing code repositories, and also widens the range of scenarios in which differentially-private mechanisms can be applied with robust privacy guarantees.

Contents

0 Introduction 1
0.1 Contributions . . . 8

1 Preliminaries 10
1.1 Differential Privacy . . . 11
1.2 Interactive Mechanisms . . . 18
1.3 Composition . . . 21
1.4 Odometers and Filters . . . 24
1.5 Generalized Probability Distance . . . 30

2 Odometers and Filters 33
2.1 Motivation . . . 33
2.2 Filters . . . 35
2.3 Odometers . . . 38
2.4 I.M.-to-I.M. Postprocessing . . . 44
2.5 Correspondence Between Odometers and Filters . . . 50

3 Concurrent Filters and Odometers for f-DP 55
3.1 f-Differential Privacy . . . 55
3.2 Concurrent Filter & Odometer for f-DP . . . 60
3.3 Sample f-DP Filter & Odometer . . . 63

4 Concurrent Filters and Odometers for Rényi Divergence-Based DP Definitions 66
4.1 Rényi and Zero-Concentrated Differential Privacy . . . 66
4.2 Concurrent Filter for Rényi-DP . . . 70
4.3 Concurrent Odometer for Rényi-DP . . . 76
4.4 Concurrent Filter & Odometer for zCDP . . . 77

5 Conclusion 78
5.1 Practical Implications . . . 79

5.2 Future Directions 79

References **84**

Listing of figures

1 An adversary $\mathcal{A}$ interacts with mechanisms $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$ in sequence. The composition of $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$ can be viewed as an overall mechanism $\mathcal{M}$. 5

1.1 Interaction between an adversary $\mathcal{A}$ and $\mathcal{M} := \mathrm{ConComp}(\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3)$, with parameters $\varepsilon_1, \varepsilon_2, \varepsilon_3$ respectively fixed in advance. 23

2.1 Concurrent composition with adaptively-chosen privacy loss parameters. Interaction between $\mathcal{A}$ and $\mathcal{M} := \mathrm{ConComp}(\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \dots)$ 34

2.2 Schematic of a concurrent approximate-DP filter. 38

2.3 Schematic of a concurrent approximate-DP odometer. 44

2.4 Person-in-the-middle mechanism. The *person-in-the-middle* $\mathcal{P}$ can communicate interactively with both adversary $\mathcal{A}$ and mechanism $\mathcal{M}$, as shown, with $\mathcal{A}$ choosing queries adaptively based on the messages it receives from $\mathcal{P}$. The message that $\mathcal{P}$ finally passes on can be thought of as the result of an interactive postprocessing. 45

3.1 Schematic of IM-to-IM postprocessing $\mathcal{P}$ for Theorem 3.2.1. $\mathcal{P}$ will only send requests to start new mechanisms to $\mathcal{F}$-*Filt*(*NIM*); for queries to existing mechanisms, $\mathcal{P}$ will use the corresponding interactive NIM-to-IM postprocessing to answer the query. 61

4.1 Schematic of IM-to-IM postprocessing $\mathcal{P}$ for Theorem 4.2.1. $\mathcal{P}$ starts $\mathcal{M}_1$ and $\mathcal{F}_{K-1}$ together following the adversary's request to start $\mathcal{M}_1$, and passes queries to mechanisms that are not to $\mathcal{M}_1$ to their corresponding mechanisms in $\mathcal{F}_{K-1}$. 74

Dedicated to my grandparents.

Acknowledgments

This thesis would not have been possible without the support of my advisor Salil Vadhan, who welcomed me into his group OpenDP with open arms and mentored me throughout my journey into differential privacy. Your patience and generosity has helped me grow both as a scientist and a human being, and made this process a deeply worthwhile one. I'm glad I also got to learn about surfing from you.

To my mentor Wanrong Zhang, who spent so much time guiding me through all aspects of computer science research and tirelessly looking over my drafts. I am infinitely grateful for our late nights together – thank you for your faith in me.

Thank you to Cynthia Dwork for being my second thesis reader, and for your course on algorithmic fairness and differential privacy that sparked my interest in this subject in the first place. I look forward to your comments.

Thank you to Samuel Haney, Michael Shoemate, and Andrew Vyrros for working through the ideas at various stages of the process, and to Grace Tian for your prior work in this area. Thank you to Annie Wu, Lindsay Froess, and Allison Choat for keeping the OpenDP and the Privacy Tools Project engines running, so that thesis writing could be a logistical breeze.

Thank you to Andrew Chen, Talia Blatt, Samar Ahmad, and Ishan Bhatt for your eyes on my drafts. It truly takes a village.

Finally, a special thank-you to Harvard Undergraduate Women in Computer Science, which provided much-needed community and encouragement throughout my undergraduate career. From WiCS, two people I endlessly look up to are Nari Johnson and Rachel Hong, whose work encouraged me to embark on computer science research in the first place.

This work was funded in part by NSF grant BCS-2218803. A preliminary version of this thesis was published at the Association of Computing Machinery's Conference on Computer and Communications Security 2023, under the title "Concurrent Composition for Interactive Differential Privacy with Adaptive Privacy-Loss Parameters."

Most people, even today, tend to think of mass surveillance in terms of content – the actual words they use when they make a phone call or write an email. The unfortunate truth, however, is that the content of our communications is rarely as revealing as its other elements – the unwritten, unspoken information that can expose the broader context and patterns of behavior.

Edward Snowden, *Permanent Record*

0

Introduction

In recent years, electronic records have become more detailed and more easily accessible than ever. The large data models and recommender systems found in social media and on video-sharing sites[35] come to mind immediately, but today's proliferation of data collection tools means that even tasks of daily life, such as shopping online, going for a health check-up, or visiting the DMV, cause us to contribute to a database in some way. The question of our times is no longer whether we can avoid having our data floating around at all, but whether we can protect our anonymity while maintaining

a reasonable level of engagement with the world.

The truth is that data does make life easier. The more exhaustive a dataset is, the more interesting, useful, or customizable results an analysis running on it can produce. For instance, a diabetes predictor model that has access to height, age, sex, weight, race, and family medical history can be much more informative than a model that only has access to height, age, and sex, because it can take more features into account. Even statistics like the population density and the demographics of a region, generated by simple aggregations of data, can help a government official decide how and in what ways to partition county resources to best serve a city.

However, the vast availability of detailed data means that analyses can reveal sensitive information such as finances, political preferences, and employment and medical history. These privacy violations carry financial, personal, social, and reputational risks. Even if a privacy violation never occurs, the threat of information leakage can erode systemic trust. For example, a person might be less likely to disclose genetic history to health practitioners, even though that would improve the care they receive, if they believe their insurance company might catch wind of it – and raise their premiums.

Any database is vulnerable to hacking, the most tremendous violation of privacy possible, but even good-faith analyses on anonymized data may still have pernicious impacts on privacy. This thesis is more concerned with the latter case.

Researchers have shown heuristic anonymization measures such as removing personally identifiable information can still fail to prevent catastrophic information leakage. Linkage attacks and differencing attacks are two such examples. In a linkage attack,[36,32] an adversary matches anonymized records in one database with de-anonymized records in another based on various attributes. For instance, a data analyst may have access to anonymized healthcare records with attributes such as gender, age, postal code, and health condition for a certain hospital. The analyst can find public voter records with name, gender, age, and postal code, and can recover who has which health condition by matching the voter records with the healthcare records. The key to this attack is that the set

of people with the same gender, age, and postal code could be very small, so matching health condition to names is not a lot of work. In a differencing attack,[9] an adversary cross-references queries of multiple databases to reveal sensitive information about an individual who was in those databases. Even aggregate statistics are enough in this case. To illustrate, suppose a city's records includes the average income of all its adult residents, the average income of all the adult residents who are members of a homeowner's association, adult population size, and homeowner's association sizes. If only one person in the city is not in a homeowner's association, then simple arithmetic over these aggregated statistics can leak that person's income. The privacy damage is less overt than the release of an immediate database, but no less troubling.

Minorities and marginalized groups especially stand to be harmed by privacy attacks, as their already-vulnerable position could make them a target for harassment and hate crimes.[1,8] However, any individual hoping to protect their sensitive data, but still wanting to benefit from the results of a data analysis, would find themselves in tough luck. The analyst thus grapples with a fundamental tension between ensuring the privacy of members of a dataset and making meaningful conclusions from it.

One framework that attempts to address this issue is differential privacy (DP), a statistical definition of privacy that attempts to quantify the impact of an algorithm on the anonymity of members of the dataset. A mechanism that is differentially private allows the analyst to gain useful information about the dataset without learning identifying information about any individual in it. The setting is a *protocol* between an *analyst* (or adversary) of arbitrary computational power and the *mechanism* it is running on a dataset. Take *mechanism* to mean any arbitrary function that could be as complicated as a neural network with hundreds of layers or as simple as a summing operation. Informally, any sequence of results to an analyst's queries for a differentially-private mechanism should be "similarly likely" with or without any individual's data. We measure this similarity by computing a statistical divergence between the distributions of the outputs of the algorithm on two

datasets that differ on one individual's data, and this divergence is the *privacy-loss parameter(s)* of the algorithm. This probability is taken over the random choices the mechanism makes at each step of computation, which we sometimes refer to as the "coins" or "coinflips" of a mechanism. In this way, any individual's presence in the dataset should only have a limited impact on the conclusions reached by a differentially-private algorithm, as bounded by the privacy parameters.

We can think of the privacy parameter as a hyperparameter that the analyst can tune. The privacy parameter takes on a different form depending on the definition of DP, which is in turn based on different notions of what it means for outputs to be "similarly likely" if we just change one individual's data between datasets. There are many such definitions, but this thesis will further explore the variants of pure-DP, [11] *f*-DP, [40,10] Rényi-DP, [31] and zero-concentrated DP. [16,5]

Simple examples of DP mechanisms often involve adding some well-calibrated amount of noise to the output of a mechanism. For example, the Laplace mechanism [9] perturbs the output of a non-differentially private function with Laplacian noise that is scaled to privacy parameter ε and the sensitivity of that function to any one person in the dataset. Another type of simple DP mechanism drawing a sample from a set of objects according to a probability scaled to how "good" it is with respect to the dataset. For example, the Exponential mechanism [29] takes in a database, a set of objects, and a score function that maps the database and set of objects to some real-valued number (a "score"). The probability that any particular object in the set is output is proportional to an exponential factor that takes into account a privacy parameter ε, the score of the object, and the sensitivity of the score function. Essentially, DP sacrifices a small amount of accuracy for a significant privacy guarantee, which comes in the form of the difficulty of deducing any individual's impact on results of the algorithm on the dataset.

DP mechanisms can either be non-interactive or interactive. DP was first defined in terms of non-interactive mechanisms. [11,12] In a non-interactive setting, the analyst collects all queries it will send to the mechanism ahead of time and receives the answers back in a one-off batch. After receiv-

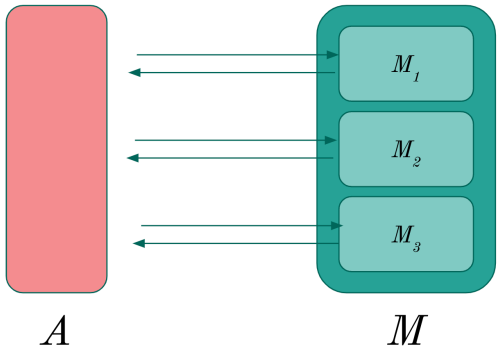


Figure 1: An adversary $\mathcal{A}$ interacts with mechanisms $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$ in sequence. The composition of $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$ can be viewed as an overall mechanism $\mathcal{M}$.

ing these answers, the analyst will engage no more with the mechanism. More recently, researchers have extended DP to interactive mechanisms.[37,38] An interactive mechanism, or I.M., allows the analyst to adaptively choose queries to ask based on the answers it has previously received. For example, suppose the query space is $\{q_1, q_2, q_3\}$. If the analyst asks q_1 first of the mechanism, and the mechanism gives an answer a_1, the analyst can then decide whether to ask q_2 or q_3 based on the value of a_1. A non-interactive interaction can be viewed as a one-off exchange between an analyst and a mechanism, while an interactive interaction can be viewed as a dialogue between the two.

We can construct more sophisticated DP algorithms by using the simpler ones as building blocks. Composition theorems provide privacy-loss guarantees for these combinations of DP mechanisms,[11] and the guarantees will vary based on the definition of DP used (pure-DP, approximate-DP, etc.).[15,16,31,10]

Composition also comes in a number of settings. In sequential composition, the analyst picks a set of mechanisms and their privacy parameters beforehand, and during the computation will interact with each mechanism in order to get an output in a sequence, as depicted in Figure 1. A twist on sequential composition is *fully-adaptive* composition, in which the analyst can adaptively choose the privacy-loss parameters of the mechanisms as it proceeds with the computation. As an example,

suppose the adversary is interacting with two non-interactive pure-DP mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$. Standard, non-adaptive composition says that if we fix ε_1 and ε_2 upfront as respective privacy parameters of the pure-DP mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively, then their composition will also be pure DP, with parameter $(\varepsilon_1 + \varepsilon_2)$. In the fully-adaptive setting, suppose the analyst doesn't yet know what privacy parameter it wants to choose for $\mathcal{M}_2$. It starts $\mathcal{M}_1$ first with parameter ε_1, and then based on the output of $\mathcal{M}_1$ on its input, decides on an ε_2 to be the privacy parameter for $\mathcal{M}_2$. The fully-adaptive composition of $\mathcal{M}_1$ and $\mathcal{M}_2$ has been shown to also be pure DP with parameter $(\varepsilon_1 + \varepsilon_2)$, meaning that the adversary's ability to choose privacy budget as it goes along does not impact the privacy guarantees of the overall computation. [11]

Interactive mechanisms allow us to work with a very general form of composition called *concurrent composition.* In this setting, the adversary can take advantage of the I.M.'s interactivity by not only adaptively asking queries of the same mechanism, but also interleaving queries between different mechanisms. Concurrent composition complicates privacy guarantee analyses. For instance, we may know the privacy guarantee of two separate interactive mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$, and if we compose them sequentially, we can hardwire $\mathcal{M}_1$'s answers into the adversary's strategy against $\mathcal{M}_2$'s. However, if the adversary can interact with $\mathcal{M}_1$ together with $\mathcal{M}_2$, the adversary can potentially leverage a privacy attack against either one of $\mathcal{M}_1$ and $\mathcal{M}_2$ based on the sequence of answers received from the other.

Researchers have found that concurrent composition theorems follow their sequential non-interactive counterparts for approximate DP, as well as some other DP variants such as *f*-DP and Rényi DP. [37,38] Essentially, in practice, no additional work needs to be done to account for the additional interactivity of concurrent composition. However, current concurrent composition theorems are all for the setting in which the privacy-loss parameters are fixed upfront. The fully-adaptive concurrent composition of mechanisms has not yet been studied, and is the work of this thesis.

We study fully-adaptive concurrent composition through the lens of *privacy odometers and fil-*

	Sequential	Concurrent
Privacy parameters pre-fixed	pure-DP,[11] approx.-DP,[11] f-DP,[10] Rényi-DP,[31] zCDP[16,5]	pure-DP,[37] approx.DP,[37] f-DP,[38] Rényi-DP,[27,38] zCDP[27]
Fully-adaptive	pure-DP,[34] approx.-DP,[34] Rényi-DP,[28] zCDP[34,41]	*Research gap*

Table 1: Table of prior work and current research gap for composition. We fill in the current research gap for fully-adaptive sequential composition of f-DP, as well as fully-adaptive concurrent composition of pure-DP, approximate-DP, f-DP, Rényi-DP, and zCDP.

ters, a particular use-case of composition with adaptive privacy-loss parameters. Privacy odometers and filters are two privacy-loss tracking primitives that allow the analyst to adaptively choose privacy-loss parameters over the course of computation. A privacy-loss *odometer* keeps track of how much privacy loss was incurred over the course of computation. Similar to checking a car odometer on a dashboard, an analyst can check in on this progress by asking the odometer a special `privacy_loss` query. A privacy-loss *filter* is an interactive mechanism that automatically halts computation on a dataset once a privacy-loss budget it is initialized with is reached.[34,28,18] Odometers and filters are themselves essentially interactive mechanisms equipped with special properties (support for a `privacy_loss` query, and a continuation rule, respectively).

As an example for the odometer setting, the analyst has started k mechanisms $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k$ with parameters $\varepsilon_1, \ldots, \varepsilon_k$. Suppose they want to check in on the privacy loss incurred to see if they want to start a $(k+1)$st mechanism, and if so, make a decision on how big its privacy-loss parameter ε_{k+1} should be. The analyst can ask the odometer a `privacy_loss` query to get this value, and then adaptively choose ε_{k+1} based off of that.

As an example for the filter setting, suppose the analyst has a preset privacy budget of ε. The adversary can use a filter to start its mechanisms, and the filter will halt if the adversary's next query to start a new mechanism would cause the privacy loss to exceed the budget.

Previous literature has focused on privacy-loss odometers and filters for the *sequential* composition of *non-interactive* mechanisms.[34,28,41] In this setting, each query is associated with a privacy loss

because each query starts a new mechanism, which has its own privacy parameter.

Privacy-loss odometers and filters for concurrent composition of interactive mechanisms has not been studied until now. This is concurrent composition with fully-adaptive privacy-loss parameters. What makes this problem unique and rich is the level of flexibility the setting allows the adversary. Not only can the adversary adaptively decide the length of its interaction with each mechanism, it can also adaptively choose the privacy-loss parameters of the next mechanism it composes. As this additional freedom may offer the adversary additional opportunities to break privacy, understanding the theoretical trappings of concurrent composition with adaptive privacy-loss parameters under multiple definitions of DP is a key step in building privacy-loss primitives that support a wider range of scenarios. We will see that results for the fully-adaptive composition of interactive mechanisms match pre-existing results for the fully-adaptive composition of non-interactive mechanisms, which implies that the additional adaptivity between choice of query and privacy-loss parameters offered by concurrent filters and odometers does not lead to privacy degradation.

0.1 Contributions

In this thesis, we demonstrate that privacy guarantees for the fully-adaptive composition of non-interactive mechanisms extend to the composition theorems for the fully-adaptive, concurrent composition of interactive mechanisms for f-DP, Rényi-DP, and zero-concentrated differential privacy (zCDP) variants. Our odometers and filters for concurrent composition also simplify existing results for concurrent composition of f-DP mechanisms, and present a new formulation of concurrent odometers and filters for Rényi-DP and zCDP.

- **Theorems for concurrent composition with adaptive privacy-loss parameters for f-DP, Rényi-DP, and zCDP.** In Chapters 3 and 4, we construct and prove the validity of concurrent odometers and filters for f-DP, Rényi-DP, and zCDP, each chosen for their com-

positional properties. We also show how to simplify previous works' results, and provide an example of an approximate-DP concurrent odometer and filter given our formulations. We find that for all of *f*-DP, Rényi-DP, and zCDP in spite of the increased flexibility, no additional privacy guarantee relaxation is necessary. Concurrency implies sequentiality, so our results are strictly stronger than previous ones, and can be used to reformulate cleaner proofs for some previously obtained theorems as well.

- **Formulation of concurrent odometer and filter primitives for generalized DP mechanisms.** In Chapter 2, we define odometers and filters for the concurrent composition of mechanisms in terms of state machines. DP mechanisms were first defined in terms of next-message functions, but the state machine reformulation allows for cleaner proofs and a more direct relationship with practical libraries.

- **Closure under interactive processing, and correspondence between odometers and filters.** Also in Chapter 2, we define I.M.-to-I.M. interactive postprocessing, and prove that I.M.-to-I.M. interactive postprocessing preserves privacy guarantees. We show that one can construct a valid filter's continuation rule from a valid odometer's privacy accumulator function, and vice versa. This is an important component of our proofs in Chapter 3 and 4.

Finally, these results simplify existing libraries. Large-scale differential privacy deployments often result in very complicated systems involving both concurrent and interactive compositions. For the libraries of OpenDP [19] and Tumult Labs [25,4] in particular, implementing some such compositions required enforcing rigid sequentiality constraints. The privacy guarantees for concurrent odometers and filters in this thesis allow the implementations to do away with these constraints.

1

Preliminaries

In this chapter, we will introduce the notation for the rest of this thesis.

Differential privacy operates on the level of databases, and so we will take a set $\mathcal{X}$ as a database space. For example, if we are working with a set of records that contains a finite and ordered n number of entries, then we would say $\mathcal{X} = \mathbb{R}^n$.

Two datasets x and x' in a database space are considered *neighboring* if they differ on the data

of one entry in the dataset. There are two kinds of differential privacy, *bounded* and *unbounded*.[23] Bounded DP fixes x and x' to be the same size, whereas unbounded DP has no such restriction.[13] The results in this thesis hold for both bounded and unbounded DP.

To understand what "neighboring" looks like for bounded DP, suppose database x and x' both contain a finite n number of real-valued entries, and on the ith row database x has entry 1 while database x' has entry 0, but all other rows are the same. We would say x and x' are neighboring. As for unbounded DP, suppose a database x comprises of dataset with a finite number n rows, with each row corresponding to an individual's data. Without loss of generality, if database x' comprises of the first $n-1$ rows of dataset x, then x and x' differ only on the nth individual's data. In this case, we would also say x and x' are neighboring.

There is some level of flexibility in the granularity of the neighboring constraint, and it will affect the sensitivity of the differential privacy algorithms defined on the graph. As another example, consider a database that takes the form of a graph. Here, two possibilities for "neighboring" datasets are two graphs that differ on one vertex, and two graphs that differ on one edge. Note that two graphs that differ on one vertex can differ up to n edges, where n is the degree of the most connected vertex. A differentially private algorithm with a given parameter will thus have a *stronger* privacy guarantee when "neighboring" is defined as a difference of a vertex, versus a difference of an edge.

1.1 Differential Privacy

Differential privacy[11] is a statistical framework to quantify the privacy loss incurred by an algorithm running on a dataset. The algorithm maps from the database space $\mathcal{X}$ to some output space $\mathcal{R}$.

Definition 1.1.1 (Differential privacy[11])**.** *A randomized mechanism* $\mathcal{M} : \mathcal{X} \to \mathcal{R}$ *is* (ε, δ)-differentially private *iff for every pair of neighboring datasets* $x, x' \in \mathcal{X}$*, and for every subset of possible*

outputs $\mathcal{S} \subseteq \mathcal{R}$,

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(\varepsilon) \cdot \Pr[\mathcal{M}(x') \in \mathcal{S}] + \delta$$

for $\varepsilon, \delta \geq 0$. *The* privacy-loss random variable *is defined as* $\mathcal{L}^{(\mathcal{S})}_{\mathcal{M}(x)||\mathcal{M}(x')} = \ln\left(\frac{\Pr[\mathcal{M}(x)\in\mathcal{S}]}{\Pr[\mathcal{M}(x')\in\mathcal{S}]}\right)$.

Intuitively, Definition 1.1.1 states that there is a very low probability that any output of $\mathcal{M}$ on dataset x will occur with significantly greater likelihood than any output of $\mathcal{M}$ on dataset x'. Therefore, the distributions of $\mathcal{M}(x)$ and $\mathcal{M}(x')$ should look very similar. The parameter ε captures the strength of this privacy guarantee. The parameter δ captures the relaxation of this privacy guarantee, i.e. some chance of privacy "leakage" wherein the distributions of $\mathcal{M}(x)$ and $\mathcal{M}(x')$ differ on some output in some significant way. When $\delta = 0$, we call this pure differential privacy (pure-DP), and just write ε-DP. When we allow $\delta > 0$, we call this approximate differential privacy, and write it as (ε, δ)-DP. [†]

In later chapters of this thesis, we will normalize the database space $\mathcal{X}$ as $\{0,1\}^*$ and the output $\mathcal{R}$ also to $\{0,1\}^*$. This representation translates to a binary string encoding, as a computer would do in practice. Note that this only allows countable $\mathcal{X}$, not $\mathbb{R}^n$; thus, we will restrict ourselves to countable $\mathcal{X}$ for the rest of this thesis. We will correspondingly also restrict our outputs to countable sets. There is a practical reason for this. Computers have finite precision, meaning that all infinite-digit floating point numbers (as are in the domain of an uncountable set, like $\mathbb{R}^n$) must be stored as finite-digit approximations. Researchers have demonstrated that it is possible to reconstruct entire datasets by examining the lower-order bits of the output of a differentially-private mechanism. [30,7] With this normalization, the distance between datasets will simply be the Hamming distance between the strings, which is the minimum number of bit flips required to change one

[†]Note: an (ε, δ)-DP mechanism can have multiple scenarios of failure. The probability δ of failure can mean that with probability $1 - \delta$, the adversary learns with certainty the data that should be protected. Alternatively, the δ parameter can mean that privacy of the mechanism degrades to some ε_1-DP with failure mode $\frac{\delta}{2}$, ε_2-DP with failure mode $\frac{\delta}{4}$, and so on. In either case, the mechanism would be overall ε-DP with probability $1 - \delta$ of failure, and both scenarios would still be (ε, δ)-DP.

string into another.

For the rest of the definitions and examples in this section, we will maintain $\mathcal{X}$ and $\mathcal{R}$ for mathematical clarity.

One can also define DP through a notion of statistical divergence between two distributions. For this, a formal notion of distance between two distributions is necessary.

Definition 1.1.2 (KL divergence[24])**.** *The* Kullback-Liebler divergence *between two probability distributions P and Q defined on the same sample space is defined as*

$$D_{KL}(P||Q) = \mathbb{E}_{x\sim P}\left[\ln \frac{P(x)}{Q(x)}\right].$$

KL divergence is asymmetric, meaning that generally, $D_{KL}(P||Q) \neq D_{KL}(Q||P)$ for some probability distributions Q and P. We typically define ε-DP in terms of max divergence, which is the worst-case analog of KL divergence. Though max divergence is also asymmetric, the symmetry is recovered by the symmetry of "neighboring datasets," as in if x, x' are neighboring, then so is x', x; it doesn't matter whether we assign a dataset x and its neighbor x' or vice versa.

Definition 1.1.3 (Max divergence and approximate max divergence[15])**.** *The* max divergence *between two random variables Y and Z taking values from the same domains is defined as*

$$D_\infty(Y||Z) = \max_{T\subseteq \text{Supp}(Y)} \ln\left(\frac{\Pr[Y\in T]}{\Pr[Z\in T]}\right).$$

The approximate max divergence *between two random variables Y and Z taking values from the same domains is defined as*

$$D_\infty^\delta(Y||Z) = \max_{T\subseteq \text{Supp}(Y):\Pr[Y\in T]\geq\delta} \ln\left(\frac{\Pr[Y\in T]-\delta}{\Pr[Z\in T]}\right).$$

Definition 1.1.4 (ε-DP and approximate-DP in terms of max divergence[15]). *A mechanism* $\mathcal{M} : \mathcal{X} \to \mathcal{R}$ *is* ε-DP *iff for every pair of neighboring datasets* $x, x' \in \mathcal{X}$,

$$D_\infty(\mathcal{M}(x)||\mathcal{M}(x')) = \max_{\mathcal{S}\subseteq\mathcal{R}} \ln\left(\frac{\Pr[\mathcal{M}(x)\in\mathcal{S}]}{\Pr[\mathcal{M}(x')\in\mathcal{S}]}\right) \leq \varepsilon.$$

A mechanism $\mathcal{M} : \mathcal{X} \to \mathcal{R}$ *is* (ε, δ)-DP *iff for every pair of neighboring datasets* $x, x' \in \mathcal{X}$,

$$D_\infty^\infty(\mathcal{M}(x)||\mathcal{M}(x')) = \max_{\mathcal{S}\subseteq\mathcal{R}} \ln\left(\frac{\Pr[\mathcal{M}(x)\in\mathcal{S}]-\delta}{\Pr[\mathcal{M}(x')\in\mathcal{S}]}\right) \leq \varepsilon.$$

Essentially, ε-DP uses max divergence to bound the privacy-loss random variable.

An example of a simple differentially-private mechanism is *randomized response*, which is a surveying technique that involves collecting binary answers from a survey population. Given some data x, each individual i in the dataset independently computes a value $q(x_i) \in \{0, 1\}$ and produces a report r_i. With probability $p > \frac{1}{2}$, the individual will report $q(x_i)$ truthfully; otherwise, they will flip the value and report $1 - q(x_i)$ instead.

Example 1.1.5 (Randomized response satisfies $\ln(\frac{p}{1-p})$-differential privacy). *Let* x *and* x' *be neighboring datasets that differ on the data of an individual* i *and let* RR *denote the randomized response mechanism. We will bound the max divergence.*

By symmetry, because each individual responds independently, we need only to consider the mechanism's output on individual i *to calculate max divergence; the rest cancel out.*

Thus,

$$D_\infty(RR(x)||RR(x')) = \ln\left(\frac{\Pr[r_i = 1|q(x_i)]}{\Pr[r_i = 1|q(x_i')]}\right) \leq \ln\left(\frac{p}{1-p}\right)$$

Some mechanisms are based off of adding noise drawn from a known statistical distribution to a function applied on a database. The Laplace mechanism is one such example, operating on a histogram representation of a dataset. For a given function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$, neighboring databases

$x, x' \in \mathbb{N}^{|\mathcal{X}|}$, and a privacy parameter ε, the Laplace mechanism will add coordinatewise noise drawn from $\text{Lap}(\frac{\Delta f}{\varepsilon})$, where Δf denotes the ℓ_1 *sensitivity* of f, i.e. $\Delta f = \max_{\text{adjacent } x, x'} \|f(x') - f(x)\|_1$ where $\|\cdot\|_1$ denotes ℓ_1 norm. For the noise mechanisms, we will normalize the sensitivity of f to 1, akin to scaling all outputs of f by Δf.

Example 1.1.6 (Laplace mechanism [13])**.** *For a function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$ of ℓ_1 sensitivity at most* 1*, and a dataset $x \in \mathcal{X}$, the* Laplace mechanism *with parameter ε is defined as $\mathcal{M}(x) = f(x) + (Y_1, \ldots, Y_k)$, where each Y_i is drawn i.i.d. from the Laplace distribution with parameter $\frac{1}{\varepsilon}$.*

Theorem 1.1.7 (Laplace mechanism satisfies ε-differential privacy [13])**.** *The Laplace mechanism with parameter ε is ε-DP.*

Proof. Fix a pair of neighboring datasets x, x', and pick a function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$. Let P denote the probability distribution of $\mathcal{M}(x)$, and Q denote the probability distribution of $\mathcal{M}(x')$. At some arbitrary point $z \in \mathbb{R}^k$, the ratio of densities is

$$\begin{aligned}
\frac{P(z)}{Q(z)} &= \prod_{i=1}^{k} \frac{\exp(-\varepsilon|f(x_i) - z_i|)}{\exp(-\varepsilon|f(x'_i) - z_i|)} \\
&= \prod_{i=1}^{k} \exp\left(\varepsilon|f(x'_i) - z_i| - \varepsilon|f(x_i) - z_i|\right) \\
&\leq \prod_{i=1}^{k} \exp\left(\varepsilon|f(x'_i) - f(x_i)|\right) && \text{triangle inequality} \\
&= \exp(\varepsilon\|f(x') - f(x)\|_1) && \text{definition of } \ell_1 \text{ distance} \\
&\leq \exp(\varepsilon) && \text{definition of } \ell_1 \text{ sensitivity, and } x, x' \text{ are neighboring}
\end{aligned}$$

as desired. By symmetry, $\frac{P(z)}{Q(z)} \geq -\varepsilon$, and so we have that the Laplace mechanism is ε-DP. □

Remark: Since we restrict mechanisms to countable inputs and outputs, we should actually use the discrete analogue of the continuous Laplace distribution. Essentially, instead of a probability

density of $\frac{\varepsilon}{2}\exp(-\varepsilon|x|)$ for $x \in \mathbb{R}$ as in the continuous Laplace case for a function of sensitivity 1, the discrete Laplace distribution has a probability density of $\frac{\exp(\varepsilon)-1}{\exp(\varepsilon)+1}\exp(-\varepsilon|x|)$ for $x \in \mathbb{Z}$.[20,2]

The Laplace mechanism satisfies ε-differential privacy in both its continuous and discrete versions. We show the continuous version here for notational simplicity, but the discrete Laplace mechanism is better to use in practice to combat privacy attacks.[6]

Another mechanism is the Gaussian mechanism,[15] which adds Gaussian noise drawn from $\mathcal{N}(0, \sigma^2)$ coordinatewise to the output of some function on a dataset, where σ^2 is a parameter of the mechanism. The Gaussian mechanism adds noise scaled to the ℓ_2 sensitivity of f, instead of the ℓ_1 sensitivity.[15]

Example 1.1.8 (Gaussian mechanism[15]). *For a function* $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$ *of* ℓ_2 *sensitivity at most* 1*, where* k *is the dimension of the function, and a dataset* x*, the* Gaussian mechanism *with parameter* σ^2 *is defined as* $\mathcal{M}(x) = f(x) + (Y_1, \ldots, Y_k)$*, where each* Y_i *is drawn i.i.d. from* $\mathcal{N}(0, \sigma^2)$*.*

The Gaussian mechanism also has a discrete analogue. Instead of having a probability density of $\frac{1}{\sigma\sqrt{2\pi}}\exp(-(x-\mu)^2/2\sigma^2)$ for $x \in \mathbb{R}$ as in the continuous Gaussian case, the discrete Gaussian distribution has a probability density of $\frac{\exp(-(x-\mu)^2/2\sigma^2)}{\sum_{y\in\mathbb{Z}}\exp(-(y-\mu)^2/2\sigma^2)}$ for $x \in \mathbb{Z}$, where μ is restricted to $\mathbb{Z}$.[6] We will discuss the Gaussian mechanism in further detail in Chapter 4.

Many other definitions of differential privacy that are based on other notions of divergence between statistical distributions have enjoyed use, due to the fact that they have better compositional properties. (Composition will be further discussed in Section 1.3.) In this thesis, we will focus on f-DP (defined in Chapter 3) and Rényi-DP and zero-concentrated differential privacy (defined in Chapter 4).

Rényi-DP and zCDP are defined in terms of Rényi divergence.

Definition 1.1.9 (Rényi divergence[33]). *The* Rényi divergence *between two probability distributions*

P and Q is defined as

$$D_\alpha(P||Q) = \frac{1}{\alpha - 1} \ln \mathbb{E}_{x \sim Q} \left[\frac{P(x)}{Q(x)} \right]^\alpha .$$

KL divergence is a special case of Rényi divergence, by taking the limit as α approaches 1.[39] Max divergence is the extreme case of Rényi divergence, by taking the limit as α approaches infinity.[39]

Differential privacy is preserved under non-interactive postprocessing, which states that any function applied to the output of a private mechanism on a dataset is also private. Intuitively, an adversary cannot increase privacy loss of a mechanism simply by toying with the mechanism's outputs, without gaining additional information about the dataset (such as having access to the output of another mechanism). Concretely, privacy preservation for an approximate-DP mechanism would look like the following.

Theorem 1.1.10 (Privacy preserved under non-interactive postprocessing for approximate-DP[11])**.** *Let $\mathcal{M} : \mathcal{X} \to \mathcal{R}$ be a randomized mechanism that is (ε, δ)-differentially private. Let $f : \mathcal{R} \to \mathcal{R}'$ be an arbitrary random mapping. Then $f \circ \mathcal{M} : \mathcal{X} \to \mathcal{R}'$ is (ε, δ)-differentially private.*

Proof. We will prove this for a deterministic mapping $f' : \mathcal{R} \to \mathcal{R}'$, because any randomized mapping can be decomposed into a convex combination of deterministic mappings. A convex combination of approximate-DP mechanisms is also approximate-DP.[15]

Fix a pair of neighboring datasets x, x', and consider any possible output set $\mathcal{S} \subseteq \mathcal{R}'$. Define another output $T \subseteq \mathcal{R}$ such that $T = \{r \in \mathcal{R} : f'(r) \in \mathcal{S}\}$. Then

$$\begin{aligned} \Pr[f'(\mathcal{M}(x)) \in \mathcal{S}] &= \Pr[\mathcal{M}(x) \in T] \\ &\leq \exp(\varepsilon) \cdot \Pr[\mathcal{M}(x') \in T] + \delta \\ &= \exp(\varepsilon) \cdot \Pr[f'(\mathcal{M}(x')) \in \mathcal{S}] + \delta \end{aligned}$$

□

1.2 Interactive Mechanisms

Mechanisms can be non-interactive or interactive. In the non-interactive case, the mechanism produces an answer to an analyst's query, and the analyst does not interact any more with the dataset. Differential privacy was initially defined in this setting, and the definitions in Section 1.1 are all in terms of non-interactive mechanisms.

In the interactive case, the analyst can ask queries adaptively of the mechanism. We call an interactive mechanism an I.M. and we define I.M.s in terms of *state machines*, which maintain their own state and compute a new state based off of the previous state and the message they receive. State machines are particularly suitable for modeling interactivity because they reflect how mechanisms are often implemented in libraries such as OpenDP.[19]

Definition 1.2.1 (Interactive mechanism[37])**.** *An* interactive mechanism*, also known as a* randomized state machine*, consists of a randomized algorithm* $\mathcal{M} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$ *that takes the current state* $s \in \{0,1\}^*$*, a query* $q \in \{0,1\}^*$*, and returns a new state* $s' \in \{0,1\}^*$ *and an answer* $a \in \{0,1\}^*$*, written* $(s', a) = \mathcal{M}(s, q)$*. When we wish to make the randomness* $r \in \{0,1\}^*$ *of* $\mathcal{M}$ *more explicit, we write* $(s', a) = \mathcal{M}(s, q; r)$*.*

An adversary is simply an interactive mechanism that is initialized with an empty string λ, instead of a dataset $x \in \{0,1\}^*$.

Definition 1.2.2 (Interaction between two interactive mechanisms[37])**.** *For two interactive algorithms* $\mathcal{M}$ *and* $\mathcal{A}$*, the* interaction *between* $\mathcal{M}$ *and* $\mathcal{A}$ *on an input* $x \in \{0,1\}^*$ *to* $\mathcal{M}$ *is the following random process (denoted* $(\mathcal{A} \leftrightarrow \mathcal{M}(x))$*):*

1. *Initialize* $s_1^{\mathcal{M}} := x$*,* $m_0 = \lambda$*,* $s_1^{\mathcal{A}} = \lambda$*, where* λ *is the empty string.*
2. *Repeat the following for* $i = 1, 2, \ldots$.

(a) If i is odd, let $(s_{i+1}^{\mathcal{M}}, m_i) = \mathcal{M}(s_i^{\mathcal{M}}, m_{i-1})$ and $s_{i+1}^{\mathcal{A}} = s_i^{\mathcal{A}}$.

(b) If i is even, let $(s_{i+1}^{\mathcal{A}}, m_i) = \mathcal{A}(s_i^{\mathcal{A}}, m_{i-1})$ and $s_{i+1}^{\mathcal{M}} = s_i^{\mathcal{M}}$.

(c) if $m_i = \texttt{halt}$, then exit loop.

So far, we have defined differential privacy in terms of non-interactive mechanisms. We can define differential privacy analogously for I.M.s, based on the *view* of the adversary, which is everything the adversary sees over the course of execution. Think of it as transcript of the interaction from the perspective of the adversary.

Definition 1.2.3 (View of an adversary in interaction with I.M.[37])**.** *Let $\mathcal{M}$ be an interactive mechanism and $\mathcal{A}$ be an adversary interacting with the mechanism. $\mathcal{A}$'s* view *of $(\mathcal{A}, \mathcal{M}(x))$ is the tuple $\texttt{View}_{\mathcal{A}}(\mathcal{A} \leftrightarrow \mathcal{M}(x)) = (r_0, m_1, r_2, m_3, r_4, \ldots)$ consisting of all the messages m_i for odd i received by $\mathcal{A}$ from $\mathcal{M}$ together with random coins r_i for even i that $\mathcal{A}$ tosses when computing m_i (i.e. for even i, $(s_{i+1}, m_i) = \mathcal{A}(s_i^{\mathcal{A}}, m_{i-1}; r_i)$).*

In further references to `View` in this thesis, we will drop the subscript $\mathcal{A}$, because we will only use `View` in reference to an adversary's view.

The view of an adversary allows us to define a notion of privacy for interactive mechanisms.

Definition 1.2.4 (Approximate-DP I.M.s[37])**.** *A randomized interactive mechanism $\mathcal{M} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$ is* (ε, δ)-differentially private *if for every pair of neighboring datasets $x, x' \in \{0,1\}^*$, every interactive adversary $\mathcal{A}$, and for every subset of possible interactive transcripts $\mathcal{S} \subseteq \{0,1\}^*$,*

$$\Pr[\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x)) \in \mathcal{S}] \leq \exp(\varepsilon) \cdot \Pr[\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x')) \in \mathcal{S}] + \delta.$$

An example of a pure-DP I.M. is the mechanism `AboveThreshold`,[14] which accepts a dataset x, a stream of queries, and a threshold T. `AboveThreshold` first adds Laplace noise to the threshold T

to produce a noised threshold $\hat{T}$. Then on each query, `AboveThreshold` adds independently drawn Laplace noise to the result of the query on x, and reports the noised value if the value exceeds the noised threshold $\hat{T}$. Otherwise, `AboveThreshold` will simply report that the value did not exceed the noised threshold. It's easy to see how this can be interactive: the adversary can adaptively choose follow-up queries based on `AboveThreshold`'s binary output. It turns out that `AboveThreshold` is ε-DP if the threshold is noised with a draw from $\text{Lap}(\frac{2}{\varepsilon})$ and the queries are noised with draws from $\text{Lap}(\frac{4}{\varepsilon})$.[14]

An example of an approximate-DP I.M. is the Sparse Vector Technique, which relies on `AboveThreshold`.[26] SVT accepts all the parameters that `AboveThreshold` takes in, plus a parameter c. Every time `AboveThreshold` reports that a query exceeds the threshold, SVT will instantiate a new instance of `AboveThreshold`, until c such instantiations have been made.[14] Dwork et al. have shown that privacy degrades only when the noised above-threshold values are reported.[14,15] Because of its interactivity and statefulness, the privacy of SVT actually degrades with the number of queries that are reported above threshold during the computation, rather than the total number of queries being asked. Hence, SVT's interactivity and statefulness are crucial to its privacy analysis.

DP mechanisms can also be interactively *processed*, which involves a randomized algorithm that engages interactively with the output(s) of a mechanism. Interactive postprocessing can transform non-interactive mechanisms into interactive ones,[37] or transform one interactive mechanism into another. The privacy of interactive postprocessing of non-interactive mechanisms is guaranteed with the data-processing inequality, which states that post-processing cannot increase information,[3] because the input of the interactive postprocessing is the single answer from a non-interactive mechanism. Because the adversary does not take more information from the dataset, it does not leak information. However, because interactive postprocessing of interactive mechanisms *might* take additional information from the dataset, due to the I.M.'s behavior, it is not immediately clear that privacy holds with the same parameters. We will study interactive postprocessing of I.M.s in more

detail in Section 2.4.

1.3 Composition

An analyst may want to interact with multiple datasets over the course of a computation, or an analyst may want to run multiple algorithms on a single dataset. Mechanism composition allows the analyst to interact with datasets in a more sophisticated way, by using simpler mechanisms as building blocks.

At the most basic level, an analyst can compose non-interactive mechanisms in sequence. Basic composition of non-interactive mechanisms finds that the privacy parameters of the overall composition scale at most linearly with the privacy parameters of each individual mechanism. [11]

Theorem 1.3.1 (Basic composition of non-interactive approximate-DP mechanisms [11,12,17])**.** *Define a sequence of non-interactive mechanisms $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k$ with $\mathcal{M}_j$ being $(\varepsilon_j, \delta_j)$-DP. Then their composition $\mathcal{M} := Comp(\mathcal{M}_1, \ldots, \mathcal{M}_k)$ is (ε, δ)-DP where $\varepsilon = \sum_{j=1}^{k} \varepsilon_j$ and $\delta = \sum_{j=1}^{k} \delta_j$, where $\mathcal{M}(x)$ returns a vector $(\mathcal{M}_1(x), \ldots, \mathcal{M}_k(x))$, with each mechanism executed on its own random coins.*

Since then, advanced composition theorems have found sublinear bounds for the composition of approximate-DP mechanisms. [15]

In fully-adaptive composition of non-interactive mechanisms, the adversary can adaptively choose the privacy parameters at the time of composition for each algorithm. This provides an added flexibility for data analysts, who may not know precisely in advance how much analysis to do on a certain dataset using a certain algorithm. For instance, an analyst may decide, after doing some exploratory analysis, that the rest of their privacy budget may be better suited for fine-grained analysis.

In some cases, an analyst may want to engage with multiple datasets and algorithms at the same time, using a message received from one algorithm to decide what question to ask of another. Essentially, given a sequence of composed interactive mechanisms, the analyst may wish to adaptively interleave queries *between* mechanisms, perhaps returning to previous mechanisms that it has interacted with. For instance, for two mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$, the analyst may want to ask a query of $\mathcal{M}_1$, then ask a query of $\mathcal{M}_2$, and return to ask a query of $\mathcal{M}_1$ based on the response of $\mathcal{M}_2$. Because I.M.s allow adaptive querying, *concurrent composition of interactive mechanisms*[37] is a natural formulation of this setting. We define concurrent composition formally below, and also depict it in Figure 1.1.

Definition 1.3.2 (Concurrent composition of I.M.s[37])**.** *Let $\mathcal{M}_1, \ldots, \mathcal{M}_k$ be interactive mechanisms that take private datasets $x_1, \ldots, x_k$ respectively. Then their concurrent composition $\mathcal{M} := ConComp(\mathcal{M}_1, \ldots, \mathcal{M}_k)$ is defined in Algorithm 1.*

Algorithm 1 Concurrent composition of interactive mechanisms

procedure $\mathcal{M}(s, m)$:
  if $m = \lambda$, where λ is the empty string **then**
    $x \leftarrow (x_1, \ldots, x_k)$
    $s \leftarrow (x, [(\mathcal{M}_1, s_1), \ldots, (\mathcal{M}_k, s_k)])$ ▷ initialize k mechanisms
    return (s, λ)
  end if
  Parse $s = (x, [(\mathcal{M}_1, s_1), \ldots, (\mathcal{M}_k, s_k)])$
  Parse m
  if $m = (j, q)$ where $j = 1, \ldots, k$ and q is a query to $\mathcal{M}_j$ **then**
    $(s'_j, m') \leftarrow \mathcal{M}_j(s_j, q)$
    $s' \leftarrow (x, [(\mathcal{M}_1, s_1), \ldots, (\mathcal{M}_j, s'_j), \ldots, (\mathcal{M}_k, s_k)])$
  else if m cannot be parsed correctly **then**
    $s' \leftarrow s, m' \leftarrow$ `invalid query`
  end if
  return (s', m')
end procedure

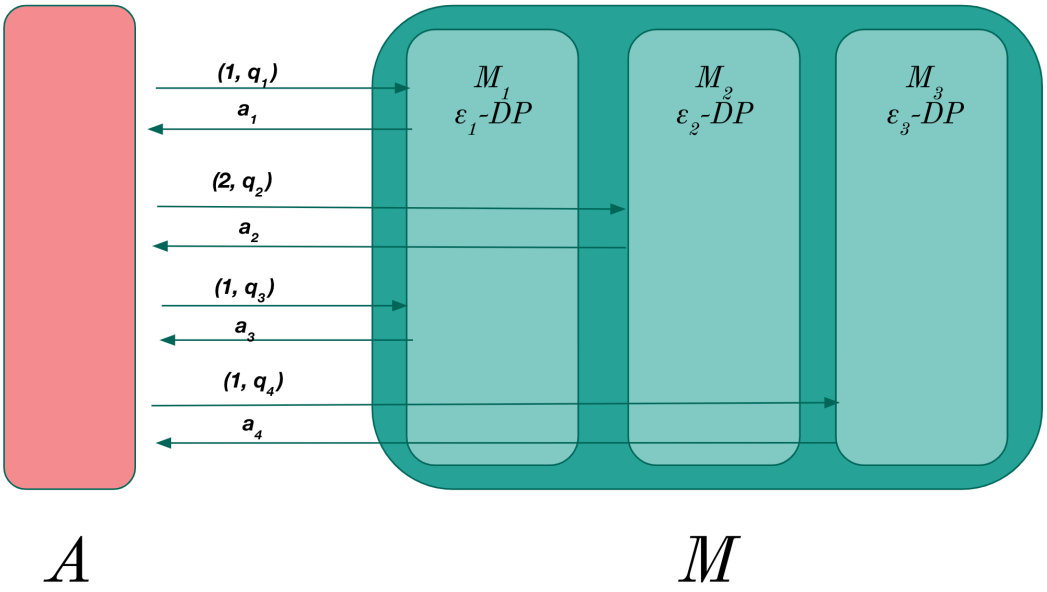


Figure 1.1: Interaction between an adversary $\mathcal{A}$ and $\mathcal{M} := \mathsf{ConComp}(\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3)$, with parameters $\varepsilon_1, \varepsilon_2, \varepsilon_3$ respectively fixed in advance.

For types of DP for which interactive mechanisms can be simulated by non-interactive mechanisms, concurrent composition theorems can follow nicely from sequential composition laws.

Theorem 1.3.3 (Concurrent composition of f-DP mechanisms[38])**.** *Let a sequence of non-interactive mechanisms $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k$ be such that $\mathcal{M}_i$ is f_i-DP, and their composition $\mathcal{M} := Comp(\mathcal{M}_1, \ldots, \mathcal{M}_k)$ is f-DP. Then the concurrent composition of a sequence of interactive mechanisms $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k$, where $\mathcal{M}_i$ is f_i-DP, denoted $ConComp(\mathcal{M}_1, \ldots, \mathcal{M}_k)$, is also f-DP.*

While there is no known simulation of interactive Rényi-DP mechanisms by non-interactive Rényi-DP mechanisms, it turns out that the privacy bounds of the concurrent composition of Rényi-DP mechanisms matches the privacy bounds for the sequential composition of Rényi-DP mechanisms.

Theorem 1.3.4 (Concurrent composition of Rényi-DP mechanisms[27,38])**.** *Fix an order $\alpha > 1$. Let*

a sequence of interactive mechanisms $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k$ be such that $\mathcal{M}_i$ is (α, ε_i)-RDP. Then their concurrent composition $\mathcal{M} := ConComp(\mathcal{M}_1, \ldots, \mathcal{M}_k)$ is $(\alpha, \sum_{i=1}^{k} \varepsilon_i)$-RDP.

All of the above concurrent composition theorems are defined for mechanisms with their privacy parameters already fixed upfront.

1.4 Odometers and Filters

Allowing full adaptivity where not only the mechanisms but also the privacy-loss parameters and the length of the composition can be chosen adaptively can lead to more efficient use of privacy resources. Privacy *odometers* and privacy *filters*, first defined by Rogers, Roth, Ullman, and Vadhan [34] for non-interactive approximate-DP mechanisms, are useful for this setting.

A *privacy filter* is a mechanism that halts computation on a dataset once a preset privacy budget is exceeded. [34] The filter is equipped with a continuation rule, which maps to a binary value either `continue` or `halt`, that ensures the interaction at each stage is differentially-private according to the privacy budget. Interaction can only continue if the continuation rule outputs `continue` at a stage. Otherwise, if the continuation rule outputs `halt`, the analyst is barred from asking the query. In practice, a `halt` can mean that filter loops until the analyst asks a query that would not violate the privacy budget, or that the filter halts to prevent any future computation on the dataset. For notational ease, we will have the filter halt in the pseudocode in this thesis. However, these scenarios bear the same privacy implications.

A *privacy odometer* is a mechanism that tracks its privacy loss over the course of computation. [34] The odometer is equipped with a privacy-loss accumulator, which is a function that outputs the privacy loss up to the round of computation in response to a special `privacy_loss` query.

In previous literature, odometers and filters have been defined in terms of the adaptive sequential composition of non-interactive mechanisms. The original odometers proposed by Rogers et al., [34]

defined for approximate-DP, requires a simultaneous guarantee bounding a privacy loss at all points *in time* when composing with adaptive privacy-loss parameters. For a $\texttt{View}$ v at a given point in time, the privacy loss is defined as $\texttt{Loss}(v) = \ln\left(\frac{\Pr[\texttt{View}(\mathcal{A}\leftrightarrow\mathcal{O}(x))=v]}{\Pr[\texttt{View}(\mathcal{A}\leftrightarrow\mathcal{O}(x'))=v]}\right)$. The odometer defined by Rogers et al.[34] for k-fold composition, a pair of datasets x, x', and a privacy-loss accumulator $\mathcal{G}$ guarantees that

$$|\texttt{Loss}(v)| > \mathcal{G}(\varepsilon_1, \delta_1, \ldots, \varepsilon_k, \delta_k)$$

occurs with probability at most δ_g, with $v \leftarrow \texttt{View}(\mathcal{A} \leftrightarrow \mathcal{O}(x))$. The final guarantee of this definition yields a form of approximate DP, because there is some δ_g possibility of failure.

Whitehouse, Ramdas, Rogers, and Wu[41] gave constructions of odometers and filters that quantitatively improve on the results presented in Rogers et al.[34], including when the composed mechanisms satisfy zCDP or Rényi DP. However, their odometer guarantee remains the same (ε, δ) of the Rogers et al. definition.

For providing Rényi-DP guarantees when composing Rényi-DP mechanisms, Feldman and Zrnic[18] constructed privacy filters. Lécuyer gave definitions and constructions of privacy odometers for Rényi-DP.[28] Our general definition of odometers follows Lécuyer's. This definition can be thought of as being in analogy to the definition of a p-value in statistics: a p-value has an interpretation if we set a significance level α before we observe our random events. If the p-value is larger than α, we reject the null hypothesis. Similarly, the privacy-loss reported by our odometer definition has an interpretation if we set a privacy-loss threshold (ε, δ) before the mechanism is executed and halt when the odometer's privacy-loss query will exceed (ε, δ). We use Lécuyer's definition because it is more natural and general, since it is applicable to all forms of DP.

To summarize, the filter emphasizes the continuation rule such that the whole mechanism satisfies the preset privacy guarantee. This is a nontrivial problem, because the expected value of the privacy loss random variable may not be equal to its expected value at a fixed time. An odometer

emphasizes a sequence of upper bounds on the privacy loss. Our definition, as we will see in Section 2.5, allows a simple conversion between the two, but previous definitions do not have this property. For example, the odometer defined by Rogers et al. [34] is a sequence of high probability point-wise bounds, and the conversion between the filter and the odometer loses some information.

In Chapter 2, we will formulate filters and odometers that support the concurrent composition of interactive mechanisms. In Chapters 3 and 4, we will extend these primitives to interactive f-DP, Rényi-DP, and zCDP mechanisms. In the next subsections we will provide a preview of our own formulations of odometers and filters by applying them to adaptive sequential composition of non-interactive mechanisms, the same setting that previous literature contends with.

1.4.1 Odometers

Definition 1.4.1 (Odometers). *An odometer* $\mathcal{O} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$ *is an I.M. that supports a specially-designated deterministic privacy-loss query, denoted* `privacy_loss`. *Upon receiving a privacy-loss query,* $\mathcal{O}$ *will output the upper bound of the privacy loss at that point in computation.*

For example, a `privacy_loss` query can be defined as all 8-bit binary strings with its most significant digit as 1. In this space, all other queries (such as queries that start a mechanism, or queries to an existing mechanism) will have its most significant digit be 0. Because privacy-loss queries are deterministic and do not change state, they are not associated with any intrinsic privacy loss.

To measure the privacy loss at each round of computation, we use the *truncated view* of an adversary $\mathcal{A}$ interacting with an odometer $\mathcal{O}$ as defined in Definition 1.4.2. Essentially, a truncated view takes in a `View` between an adversary and a mechanism as well as a privacy-loss parameter, and truncates the `View` at the point of greatest privacy loss that does not exceed the requirement set by the privacy-loss parameter.

Definition 1.4.2 (Truncated view between adversary $\mathcal{A}$ with approximate-DP odometer $\mathcal{O}$ at (ε, δ)[28])**.** *Given a privacy-loss parameter (ε, δ), an adversary $\mathcal{A}$, and approximate-DP odometer $\mathcal{O}$, $\texttt{Trunc}_{(\varepsilon,\delta)}(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{O}(x))$ is constructed as follows:*

1. *Initialize $s_1^{\mathcal{M}} := x$, $m_0 = \lambda$, $s_1^{\mathcal{A}} = \lambda$, where λ is the empty string.*
2. *Repeat the following for $i = 1, 2, \ldots$.*
   (a) *If i is odd, let $(s_{i+1}^{\mathcal{M}}, a_i) := \mathcal{O}(s_i^{\mathcal{M}}, q_{i-1})$, and let $(s_i^{\mathcal{M}}, (\varepsilon_i, \delta_i)) = \mathcal{O}(s_i^{\mathcal{M}}, \texttt{privacy_loss})$, $s_{i+1}^{\mathcal{A}} = s_i^{\mathcal{A}}$.*
   (b) *If i is even, let $(s_{i+1}^{\mathcal{A}}, q_i) := \mathcal{A}(s_i^{\mathcal{A}}, a_{i-1}; r_i^{\mathcal{A}})$ and $s_{i+1}^{\mathcal{M}} = s_i^{\mathcal{M}}$.*
   (c) *If $\varepsilon_i \geq \varepsilon$ or $\delta_i \geq \delta$, then exit loop.*
   (d) *If $s_{i+1}^{\mathcal{M}}$ or $s_{i+1}^{\mathcal{A}} = \texttt{Halt}$, then exit loop.*
3. *Return $\texttt{Trunc}_{(\varepsilon,\delta)}(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{O}(x)) = (r_0^{\mathcal{A}}, a_1, \ldots, r_{i-2}^{\mathcal{A}}, a_{i-1})$.*

Using truncated view for analyzing odometers is first introduced by Lécuyer[28] for Rényi-DP. In this thesis, we will extend their definition to a general notion of truncated view, which allows us to quantify privacy loss with all other DP variants, where we can simply replace (ε, δ) with other privacy-loss parameters for other DP variants.

The privacy guarantee of an odometer is defined as measuring the closeness between the distributions of the truncated views of an adversary.

Example 1.4.3 (Valid approximate-DP odometer for non-interactive mechanisms)**.** *We say $\mathcal{O}$ is* a valid approximate-DP odometer *if for every pair of (ε, δ), every adversary $\mathcal{A}$, and every pair of adjacent datasets x, x',*

$$D(\texttt{Trunc}_{(\varepsilon,\delta)}(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{O}(x)))||\texttt{Trunc}_{(\varepsilon,\delta)}(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{O}(x')))) \preceq (\varepsilon, \delta),$$

where $\preceq$ *is a partial order and* $(\varepsilon_1, \delta_1) \preceq (\varepsilon_1, \delta_1)$ *iff* $\varepsilon_1 \leq \varepsilon_2$ *and* $\delta_1 \leq \delta_2$.

Odometers are typically constructed with a *privacy-loss accumulator* $\mathcal{G}$ that composes the privacy-loss parameters of certain individual queries to the odometer, which are typically interactive mechanisms. In this thesis, we will exclusively study such odometers, which we term $\mathcal{G}$-odometers. However, keep in mind that a more general odometer need not be based on a privacy-loss accumulator.

Example 1.4.4 ($\mathcal{G}$-odometer for composition of non-interactive approximate-DP mechanisms ($\mathcal{G}$-*Odom(NIM)*))**.** *Let* $\mathcal{G}$ *be a privacy-loss accumulator that takes a sequence of privacy-loss parameters* $(\varepsilon_1, \delta_1), \ldots, (\varepsilon_k, \delta_k)$ *for any* $k = 1, 2, \ldots$, *and maps to a global privacy loss* (ε, δ), *so* $\mathcal{G} : \mathbb{R}^k_{\geq 0} \times \mathbb{R}^k_{\geq 0} \to \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0}$. *A* $\mathcal{G}$*-odometer for composition of non-interactive mechanisms is denoted as* $\mathcal{G}$*-Odom(NIM). At the* $(k+1)$*st round,* $\mathcal{G}$*-Odom*$(NIM)(s, m)$ *is executed as follows:*

1. *If* $m = \lambda$, *initialize* $s = (x, [])$, *where* $[]$ *is an empty list, and return* (s, λ)
2. *Parse* $s = (x, [(\mathcal{M}_1, (\varepsilon_1, \delta_1)), \ldots, (\mathcal{M}_k, (\varepsilon_k, \delta_k))])$
3. *If* $m = (\mathcal{M}_{k+1}, (\varepsilon_{k+1}, \delta_{k+1}))$ *and* $\mathcal{M}_{k+1}$ *is* $(\varepsilon_{k+1}, \delta_{k+1})$*-DP:*
   (a) *Let* $s' = (x, [(\mathcal{M}_1, (\varepsilon_1, \delta_1)), \ldots, (\mathcal{M}_{k+1}, (\varepsilon_{k+1}, \delta_{k+1}))])$
   (b) *Let* $m' = \mathcal{M}_{k+1}(x)$
4. *If* $m =$ `privacy_loss`, *then let* $s' = s$ *and* $m' = \mathcal{G}((\varepsilon_1, \delta_1), \ldots, (\varepsilon_k, \delta_k))$
5. *Return* (s', m')

Specifically for odometers equipped with privacy-loss accumulators, then, the privacy-loss accumulator must output privacy loss bounds at any stage of computation that reflect the divergence between the views at that stage.

Example 1.4.5 (Valid approximate-DP privacy-loss accumulator for non-interactive mechanisms)**.** *We say $\mathcal{G}$ is a* valid approximate-DP NIM-privacy-loss accumulator *if $\mathcal{G}$-Odom(NIM) is a valid approximate-DP odometer.*

1.4.2 Filters

Filters are I.M.s equipped with continuation rules, which take in a privacy budget set by the adversary and a sequence of privacy parameters, and maps to a binary decision whether to stop or halt the mechanism.

Example 1.4.6 ($\mathcal{F}$-filtered composition of non-interactive (ε, δ)-DP mechanisms ($\mathcal{F}$*-Filt(NIM)*)**.** *Let $\mathcal{F}$ be a continuation rule that takes in a sequence of privacy-loss parameters $(\varepsilon_1, \delta_1), \ldots,$ and a target privacy-loss budget (ε, δ), and maps to a binary decision: $\mathcal{F} : \mathbb{R}^*_{\geq 0} \times \mathbb{R}^*_{\geq 0} \times \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \{0, 1\}$, where 1 means* `continue` *and 0 means* `halt`*, and $\mathbb{R}^*_{\geq 0} = \cup_{k=0}^{\infty} \mathbb{R}^k$. The $\mathcal{F}$-*filtered composition of non-interactive mechanisms*, denoted as $\mathcal{F}$-Filt(NIM), is an interactive mechanism. At the $(k+1)$st round, $\mathcal{F}(\cdot; (\varepsilon, \delta))$-Filt$(NIM)(s, m)$ is executed as follows:*

1. *If $m = \lambda$, initialize $s = (s, [])$, where $[]$ is an empty list. Return (s, λ), where λ is an empty string.*
2. *Parse $s = (x, [(\mathcal{M}_1, (\varepsilon_1, \delta_1)), \ldots, (\mathcal{M}_{k-1}, (\varepsilon_k, \delta_k))])$*
3. *If $\mathcal{M}_{k+1}$ is $(\varepsilon_{k+1}, \delta_{k+1})$-DP and $\mathcal{F}((\varepsilon_1, \delta_1), \ldots, (\varepsilon_{k+1}, \delta_{k+1}); (\varepsilon, \delta)) = 1$:*
   (a) *Let $s' = (x, [(\mathcal{M}_1, (\varepsilon_1, \delta_1)), \ldots, (\mathcal{M}_{k+1}, (\varepsilon_{k+1}, \delta_{k+1}))])$*
   (b) *Let $m' = \mathcal{M}_{k+1}(x)$*
4. *Else, let $s' =$* `Halt`*, and let m' store the reason for halting*
5. *Return (s', m')*

Note that at each round k of computation, whether the mechanism $\mathcal{M}_k$ is $(\varepsilon_k, \delta_k)$-DP is something the implementation needs to be able to verify, e.g. by having verified privacy-loss parameters attached to every mechanism.

Example 1.4.7 (Valid approximate-DP filter for non-interactive mechanisms)**.** *We say a continuation rule $\mathcal{F}$ is a* valid (ε, δ)-DP NIM-filter for non-interactive mechanisms *if for every pair of (ε, δ), $\mathcal{F}(\cdot; (\varepsilon, \delta))$-Filt(NIM)$(\cdot)$ is an (ε, δ)-DP interactive mechanism.*

1.5 Generalized Probability Distance

Given so many DP variants, Vadhan and Zhang[38] define a generalized notion of probability distance, termed $\mathcal{D}$ DP, to deal with DP variants at a higher level of abstraction. Distances between probability distributions in the $\mathcal{D}$ DP framework is defined in terms of partially-ordered sets.

Definition 1.5.1 (Generalized probability distance[38])**.** *A* generalized probability distance measure *is a tuple $(\mathcal{D}, \preceq, D)$ such that*

1. *$(\mathcal{D}, \preceq)$ is a partially-ordered set (poset).*
2. *D is a mapping that takes any two random variables X, X' over the same measurable space to an element $D(X, X')$ of $\mathcal{D}$.*
3. *(Postprocessing.) The generalized distance mapping D is closed under postprocessing, meaning that for every measurable function g, $D(g(X), g(X')) \preceq D(X, X')$.*
4. *(Joint Convexity.) Suppose we have a collection of random variables $(X_i, X_i')_{i \in \mathcal{I}}$ and a random variable I distributed on $\mathcal{I}$. If $D(X_i, X_i') \preceq d$ for all $i \in \mathcal{I}$, then $D(X_I, X_I') \preceq d$.*

To illustrate, consider ε-DP.

Example 1.5.2 (ε-DP in the generalized probability distance framework)**.** *The partially-ordered set* $(\mathcal{D}, \preceq)$ *is* $(\mathbb{R}^{\geq 0} \cup \{\infty\}, \leq)$*. The mapping is the max divergence as defined in Definition 1.1.4.*

Proof. The domain of possible ε is the non-negative real numbers up to infinity. A smaller ε is a stronger privacy guarantee, so the ordering is captured by $\leq$ over this domain.

Max divergence, defined in Definition 1.1.4, is closed under post-processing due to the data-processing inequality. Due to the following lemma, max divergence also satisfies joint convexity:

Lemma 1.5.3 (Lemma that implies joint convexity of max divergence[39])**.** *For every two pairs of probability distributions* (P_0, Q_0) *and* (P_1, Q_1)*, and every* $\lambda \in (0, 1)$*,*

$$D_\infty((1-\lambda)P_0 + \lambda P_1 || (1-\lambda)Q_0 + Q_1) \leq \max\{D_\infty(P_0||Q_0), D_\infty(P_1||Q_1)\}.$$

□

In Chapters 3 and 4, we will present f-DP and Rényi-DP, respectively, in this generalized probability framework.

Now we can introduce a notion of a $\mathcal{D}$ DP mechanism with a specific privacy loss parameter.

Definition 1.5.4 (d-$\mathcal{D}$ DP mechanisms[38])**.** *Let* $(\mathcal{D}, \preceq, D)$ *be a generalized probability distance. For* $d \in \mathcal{D}$*, we say an I.M.* $\mathcal{M}$ *is* d-$\mathcal{D}$ *DP if for every adversary* $\mathcal{A}$ *and every pair of datasets* x, x'*,*

$$D(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x)) || \texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x'))) \preceq d.$$

The joint convexity of $(\mathcal{D}, \preceq, D)$ implies that it suffices to consider only deterministic adversary algorithms in our proofs, as we can always generate a deterministic adversary from a randomized adversary by fixing the coin tosses of the randomized adversary. To explain, consider a randomized adversary $\mathcal{A}$ interacting with a d-$\mathcal{D}$ DP mechanism $\mathcal{M}$ on dataset x and an adjacent dataset x'. Let

I be the random variable that captures the coin tosses of $\mathcal{A}$, and let X_I denote the view of the adversary that corresponds to some random coin toss. Let $i \in \mathcal{I}$, where $\mathcal{I} = \text{Supp}(I)$, be a crystallization of I, analogous to fixing the coin tosses of the randomized adversary, i.e. $i := (r_1, r_2, r_3, \dots)$. Then we can write $X_I := \texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x)) = (r_1, a_1, r_2, a_2, \dots)$. Similarly, $X'_I := \texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x')) = (r_1, a'_1, r_2, a'_2, \dots)$. Because $\mathcal{M}$ is d-$\mathcal{D}$ DP, $D(X_i, X'_i) \preceq d$ for any $i \in \mathcal{I}$. Point 4 of Definition 1.5.1 allows us to conclude that $D(X_I, X'_I) \preceq d$. Therefore, knowing that $\mathcal{M}$ is d-$\mathcal{D}$ DP, where $(\mathcal{D}, \preceq, D)$ is jointly convex, allows us to consider an adversary acting on any *crystallization* of $\mathcal{A}$'s coin tosses as a proxy for $\mathcal{A}$ itself, as d would upper-bound the divergence between neighboring datasets of any such crystallization. Using deterministic adversaries will simplify our proofs, because we will not need to consider adversary randomness in tandem with the randomness of $\mathcal{M}$.

Lemma 1.5.5 (Deterministic adversaries[38]). *An interactive mechanism $\mathcal{M}$ is $\mathcal{D}$ DP if and only if for every pair of neighboring datasets x, x' and every deterministic adversary $\mathcal{A}$, $D(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x)) || \texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x'))) \preceq d$.*

With this lemma, in future references to adversaries, we will presume the adversary is deterministic.

In the rest of this thesis, we will use the notation $\mathcal{D}^* := \bigcup_{k=0}^{\infty} \mathcal{D}^k$.

2

Odometers and Filters

2.1 Motivation

It is not immediately clear if the ability to adaptively partition a budget across multiple *interactive* mechanisms might open up the opportunity for a privacy attack. The setting in which an adversary can adaptively choose privacy parameters as they go along, where adaptivity comes not only *between* an adversary and a particular I.M., but also in the adversary's interaction with *multiple* I.M.s, mo-

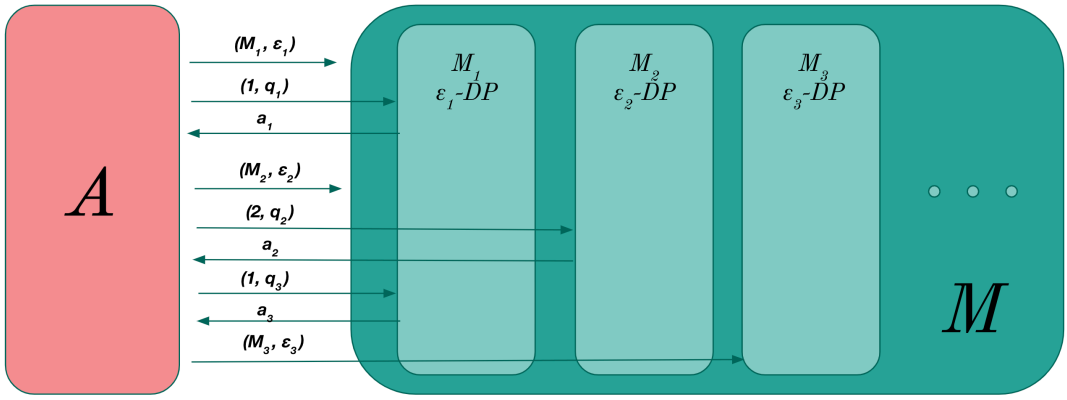


Figure 2.1: Concurrent composition with adaptively-chosen privacy loss parameters. Interaction between $\mathcal{A}$ and $\mathcal{M} := \mathsf{ConComp}(\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \dots)$

tivates our study. We will formulate privacy filters and odometers for the concurrent composition case, when privacy-loss parameters of the concurrently composed I.M.s are chosen adaptively.

Figure 2.1 further illustrates the setting we are interested in, concurrent composition with adaptive privacy-loss parameters. This figure represents the following interaction: the adversary starts $\mathcal{M}_1$ with parameter ε_1, queries $\mathcal{M}_1$ with q_1, and gets an answer. Based on this answer, the adversary then adaptively starts $\mathcal{M}_2$ with parameter ε_2, queries it with q_2, and gets an answer. The adversary then decides to query $\mathcal{M}_1$ with query q_3, and gets an answer from $\mathcal{M}_1$. Based on this answer, the adversary decides to start $\mathcal{M}_3$ with parameter ε_3. The adversary can continue to interact with the mechanisms it has already started, or start yet more.

In this chapter, we will first formulate filters for adaptive composition of non-interactive and interactive mechanisms. Then we will formulate odometers for the same, and prove that we can easily convert a filter into an odometer and vice versa without any privacy loss.

2.2 Filters

With the framework given in Definition 1.5.1, we now define privacy filters in terms of our generalized probability distances.

Definition 2.2.1 ($\mathcal{F}$-filtered composition of $\mathcal{D}$ DP non-interactive mechanisms ($\mathcal{F}$-*Filt*(NIM))**.** *For a continuation rule $\mathcal{F}(\cdot; d) : \mathcal{D}^* \times \mathcal{D} \to \mathcal{D}$, the $\mathcal{F}$-filtered composition of $\mathcal{D}$ DP interactive mechanisms is an interactive mechanism executed in Algorithm 2.*

We allow continuation rules to have a finite cap K on the number of privacy-loss parameters they can accommodate. We denote this as $\mathcal{F}_K(\cdot; d) : \mathcal{D}^{\leq K} \times \mathcal{D} \to \mathcal{D}$, where d is the privacy-loss budget. When K is finite, a filter built from $\mathcal{F}$ can only accommodate up to K mechanisms, provided that privacy loss is within budget. Notationally, if $K = \infty$ or K is not specified, the implication is that the number of mechanisms $\mathcal{F}$ can accommodate is technically unbounded.

In the $\mathcal{F}$-filtered composition of $\mathcal{D}$ DP non-interactive mechanisms, a sequence of non-interactive mechanisms are adaptively composed in real time, provided that their composition does not exceed some given privacy budget d (which $\mathcal{F}$ ensures).

We similarly define $\mathcal{F}$-filtered composition for $\mathcal{D}$ DP interactive mechanisms.

Definition 2.2.2 ($\mathcal{F}$-filtered concurrent composition of $\mathcal{D}$ DP interactive mechanisms)**.** *The $\mathcal{F}$-filtered concurrent composition of $\mathcal{D}$ DP interactive mechanisms, denoted as $\mathcal{F}$-FiltCon(IM), is an interactive mechanism as executed in Algorithm 3.*

Figure 2.2 illustrates an example of an interaction between an adversary and a concurrent approximate DP filter. In the interaction represented, the adversary first attempts to start mechanism $\mathcal{M}_1$ with parameter ε_1. When the filter verifies that this is within the privacy budget, the adversary can then freely query $\mathcal{M}_1$ with query q_1 and receive an answer. Next, the adversary does the same with $\mathcal{M}_2$ with parameter ε_2. This interaction can continue until the adversary attempts to start

Algorithm 2 $\mathcal{F}$-filtered composition of $\mathcal{D}$ DP non-interactive mechanisms ($\mathcal{F}$-*Filt*(*NIM*)), for $\mathcal{F} : \mathcal{D}^{\leq K} \to \mathcal{D}$, where K could be ∞.

procedure $\mathcal{F}(\cdot; d)$-*Filt*(*NIM*)(s, m):
  if $m = \lambda$ **then** ▷ initialize filter
    $s' \leftarrow (s, [])$, where [] is an empty list
    return (s', λ)
  end if
  Parse s as $s = (x, [(\mathcal{M}_1, d_1), (\mathcal{M}_2, d_2), \ldots, (\mathcal{M}_k, d_k)])$
  Parse m
  if $m = (\mathcal{M}', d')$ **then**
    if $k = K$ **then** ▷ already at mechanism limit
      $s' \leftarrow$ `Halt`, $m' \leftarrow$ `at mechanism count limit`
    else if $\mathcal{M}'$ is d'-$\mathcal{D}$ DP **then**
      $\mathcal{M}_{k+1} \leftarrow \mathcal{M}'$
      $d_{k+1} \leftarrow d'$
      if $\mathcal{F}(d_1, \ldots, d_{k+1}) \neq 1$ **then**
        $m' \leftarrow$ `insufficient budget`, $s' \leftarrow$ `Halt`
      else
        Random sample $r_{k+1} :=$ coin toss for $\mathcal{M}_{k+1}$
        $s' \leftarrow (x, [(\mathcal{M}_1, d_1), \ldots, (\mathcal{M}_{k+1}, d_{k+1})])$
        $m' \leftarrow \mathcal{M}_{k+1}(x; r_{k+1})$
      end if
    end if
  else if m cannot be parsed correctly **then**
    $s' \leftarrow s$, $m' \leftarrow$ `invalid query`
  end if
  return (s', m')
end procedure

Algorithm 3 $\mathcal{F}$-filtered concurrent composition of $\mathcal{D}$ DP interactive mechanisms ($\mathcal{F}$-*FiltCon*(IM)), for $\mathcal{F} : \mathcal{D}^{\leq K} \to \mathcal{D}$, where K could be ∞.

procedure $\mathcal{F}(\cdot; d)$-*FiltCon*(IM)(s, m):
  if $m = \lambda$ **then** ▷ initialize filter
    $s' \leftarrow (s, [])$, where [] is an empty list
    return (s', λ)
  end if
  Parse s as $s = (x, [(\mathcal{M}_1, d_1, s_1), \ldots, (\mathcal{M}_k, d_k, s_k)])$
  Parse m
  if $m = (\mathcal{M}', d')$ **then**
    if $k = K$ **then** ▷ already at mechanism limit
      $s' \leftarrow$ `Halt`, $m' \leftarrow$ `at mechanism count limit`
    else if $\mathcal{M}'$ is d'-$\mathcal{D}$ DP **then**
      $\mathcal{M}_{k+1} \leftarrow \mathcal{M}'$
      $d_{k+1} \leftarrow d'$
      if $\mathcal{F}(d_1, \ldots, d_{k+1}) \neq 1$ **then**
        $s' \leftarrow$ `Halt`, $m' \leftarrow$ `insufficient budget`
      else
        $(s_{k+1}, m) \leftarrow \mathcal{M}_{k+1}(x, \lambda)$
        $s' \leftarrow (x, d, [(\mathcal{M}_1, d_1, s_1), \ldots, (\mathcal{M}_{k+1}, d_{k+1}, s_{k+1})])$
        $m' \leftarrow$ `yes`
      end if
    end if
  else if $m = (j, q)$ where $j = 1, \ldots, k$ and q is a query to $\mathcal{M}_j$ **then**
    $(s'_j, m') \leftarrow \mathcal{M}_j(s_j, q)$
    $s' \leftarrow (x, d, [(\mathcal{M}_1, d_1, s_1), \ldots, (\mathcal{M}_j, d_j, s'_j), \ldots, (\mathcal{M}_k, d_k, s_k)])$
  else if m_i cannot be parsed correctly **then**
    $s' \leftarrow s$, $m' \leftarrow$ `invalid query`
  end if
  return (s', m')
end procedure

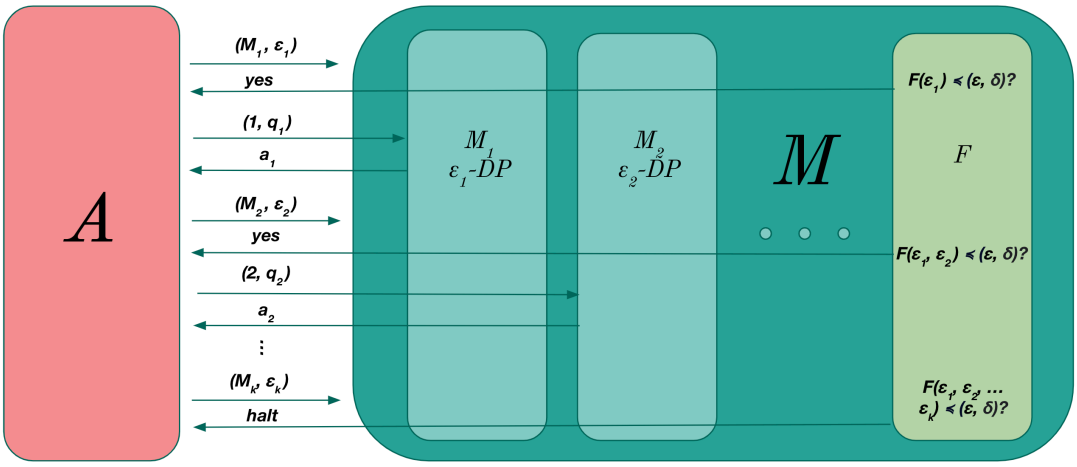


Figure 2.2: Schematic of a concurrent approximate-DP filter.

$\mathcal{M}_k$ with parameter ε_k, which in this illustration would cause the privacy loss to exceed the privacy budget. As a result, the filter returns a halt. Up until that point, the interaction would then be like $\text{ConComp}(\mathcal{M}_1, \ldots, \mathcal{M}_{k-1})$ but with adaptively-chosen privacy-loss parameters.

Important to note is that if the adversary *hasn't* yet sent a query to start $\mathcal{M}_2$ and received confirmation that it can do so, it cannot query $\mathcal{M}_2$.

Definition 2.2.3 (Valid $\mathcal{D}$ DP filter for interactive mechanisms)**.** *Let $\mathcal{F} : \mathcal{D}^* \times \mathcal{D} \to \{0, 1\}$ be a continuation rule. $\mathcal{F}$ is a valid $\mathcal{D}$ DP concurrent* IM-filter *if for every $d \in \mathcal{D}$, $\mathcal{F}(\cdot; d)$-FiltCon(IM) is a d-$\mathcal{D}$ DP interactive mechanism.*

In other words, a concurrent privacy filter is a continuation rule for which Definition 2.2.2 holds for all distances $d \in \mathcal{D}$ in the concurrent composition of interactive mechanisms.

2.3 Odometers

We can similarly define privacy odometers in terms of generalized probability distances.

Definition 2.3.1 ($\mathcal{G}$-odometer for composition of non-interactive mechanisms ($\mathcal{G}$-*Odom*(NIM))**.**

A $\mathcal{G}$-odometer for the composition of non-interactive $\mathcal{D}$ DP mechanisms, denoted as $\mathcal{G}$-Odom(NIM), is executed as in Algorithm 4.

Algorithm 4 $\mathcal{G}$-odometer for composition of non-interactive mechanisms ($\mathcal{G}$-*Odom*(NIM)) .

procedure $\mathcal{G}$-*Odom*(NIM)(s, m):
  if $m = \lambda$ **then**
    $s' \leftarrow (s, [])$, where [] is an empty list
    return (s', λ)
  end if
  Parse s as $s = (x, [(\mathcal{M}_1, d_1), (\mathcal{M}_2, d_2), \ldots, (\mathcal{M}_k, d_k)])$
  Parse m
  if $m = (\mathcal{M}', d')$ **then**
    if $\mathcal{M}'$ is d'-$\mathcal{D}$ DP **then**
      $\mathcal{M}_{k+1} \leftarrow \mathcal{M}'$
      $d_{k+1} \leftarrow d'$
      $s' \leftarrow (x, [(\mathcal{M}_1, d_1), \ldots, (\mathcal{M}_{k+1}, d_{k+1})])$
      Random sample r_{k+1} := coin toss for $\mathcal{M}_{k+1}$
      $m' \leftarrow \mathcal{M}_{k+1}(x; r_{k+1})$
    else
      $s' \leftarrow s, m' \leftarrow$ `Divergence cannot be parsed`
    end if
  else if $m =$ `privacy_loss` **then**
    $s' \leftarrow s, m' \leftarrow \mathcal{G}(d_1, \ldots, d_k)$
  else if m cannot be parsed correctly **then**
    $s' \leftarrow s, m' \leftarrow$ `invalid query`
  end if
  return (s', m')
end procedure

Our construction of odometers have a specially-designated `privacy_loss` query, which outputs a conservative bound on the privacy loss of the interaction up to that point.

Before formalizing the privacy guarantee of odometers, we first define a *truncated view* for gen-

eralized probability distances. This definition generalizes Lécuyer's definition,[28] which was for Rényi-DP.

Definition 2.3.2 (Truncated view for odometers[28])**.** *For a deterministic adversary $\mathcal{A}$ and odometer $\mathcal{O}$ on dataset x, the* truncated view

$$\texttt{Trunc}_d(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{O}(x)))$$

is the return value of Algorithm 5, including the randomness and query answers.

Algorithm 5 Truncated view of adversary interacting with odometer $\mathcal{O}$ given input d

procedure TruncView($\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{O}(x))$)
  Initialize $s_1^{\mathcal{M}} \leftarrow x, m_0 \leftarrow \lambda, s_1^{\mathcal{A}} \leftarrow \lambda$
  for $i = 1, 2, \ldots$ **do**
    if i is odd **then**
      $(s_{i+1}^{\mathcal{M}}, a_i) \leftarrow \mathcal{O}(s_i^{\mathcal{M}}, q_{i-1})$
      $(s_i^{\mathcal{M}}, d_i) = \mathcal{O}(s_i^{\mathcal{M}}, \texttt{privacy_loss})$
      $s_{i+1}^{\mathcal{A}} \leftarrow s_i^{\mathcal{A}}$
      if $d_i \not\preceq d$ **then**
        Exit loop
      end if
    else if i is even **then**
      $(s_{i+1}^{\mathcal{A}}, q_i) \leftarrow \mathcal{A}(s_i^{\mathcal{A}}, a_{i-1}; r_i^{\mathcal{A}})$
      $s_{i+1}^{\mathcal{M}} \leftarrow s_i^{\mathcal{M}}$
    end if
    if $s_{i+1}^{\mathcal{M}} = \texttt{Halt}$ or $s_{i+1}^{\mathcal{A}} = \texttt{Halt}$ **then**
      Exit loop
    end if
  end for
  return $\texttt{Trunc}_d(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{O}(x))) = (r_0^{\mathcal{A}}, a_1, r_2^{\mathcal{A}}, a_3, \ldots, r_{i-2}^{\mathcal{A}}, a_{i-1})$
end procedure

Essentially, for a given probability distance d, Algorithm 5 cuts off the adversary's view of the interaction with an odometer up to the last answer for which the privacy-loss accumulator's output

would be ordered below d.

We define a *valid* $\mathcal{D}$ DP odometer based on truncated view.

Definition 2.3.3 (Valid $\mathcal{D}$-DP odometer)**.** *$\mathcal{O}$ is a* valid $\mathcal{D}$ DP odometer *if for every $d \in \mathcal{D}$, adversary $\mathcal{A}$, and every pair of adjacent datasets x, x',*

$$D(\texttt{Trunc}_d(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{O}(x)))||\texttt{Trunc}_d(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{O}(x')))) \preceq d$$

Definition 2.3.4 (Valid $\mathcal{D}$-DP privacy-loss accumulator for non-interactive mechanisms)**.** *$\mathcal{G} : \mathcal{D}^* \to \mathcal{D}$ is a valid $\mathcal{D}$ DP non-interactive-privacy-loss accumulator (NIM-privacy-loss-accumulator) if for every $d \in \mathcal{D}$, $\mathcal{G}(\cdot)$-Filt(NIM) is a valid d-$\mathcal{D}$ DP odometer.*

In ther words, the privacy loss output by a $\mathcal{D}$-DP privacy-loss accumulator at some point in computation should be an upper bound on the actual divergence between adjacent views at that point.

It is helpful to imagine what privacy loss would look like for concurrent odometers before we define them. In sequential composition, the adversary's next query was a new mechanism, so defining privacy loss in sequential composition is the same as defining a privacy loss for each query. However, this gets more complicated with concurrent odometers. For example, as we discussed with Sparse Vector Technique in Chapter 1, one can use SVT so that each "above threshold" response reveals some information about the dataset, while each "not above threshold" response does not, meaning that a privacy guarantee associated with each query is variable and dependent on the mechanism's randomness, which is correlated across all queries. [14,21,26] Hence, articulating where and how privacy loss accumulates requires a level of care.

To address this issue for concurrent odometers, we let the privacy-loss budget to be paid upfront at the start of each additional mechanism. For example, suppose an adversary starts two Sparse Vector Techniques that it interacts with concurrently. If the privacy parameter of the first technique is ε_1 and the privacy parameter of the second is ε_2, a valid odometer may respond $\varepsilon_1 + \varepsilon_2$ in response to

a privacy-loss query after the second SVT has been started, even if only negative answers have been given in the history of the computation (as in, there has been no additional information revealed about the dataset). Also, so long as the adversary does not start a third mechanism, the odometer will always respond $\varepsilon_1 + \varepsilon_2$ to a privacy-loss query. This way, the odometer would always output the *upper bound* of the privacy loss at any given time period. Because different mechanisms will have different privacy losses, designing `privacy_loss` queries this way also allows our odometer primitive to avoid being mechanism or definition dependent.

We now define concurrent privacy-loss odometers and privacy-loss accumulators.

Definition 2.3.5 ($\mathcal{G}$-odometer for concurrent composition of interactive mechanisms ($\mathcal{G}$-*OdomCon*(IM)))**.** *A $\mathcal{G}$-odometer for the concurrent composition of interactive mechanisms, denoted as $\mathcal{G}$-OdomCon(IM), is executed as in Algorithm 6.*

Figure 2.3 depicts an example interaction between an adversary and a privacy-loss odometer built from a privacy-loss accumulator. In this example, the adversary starts $\mathcal{M}_1$ with parameter ε_1 and then queries $\mathcal{M}_1$ with q_1. Then, adaptively, it starts $\mathcal{M}_2$ with parameter ε_2 and queries $\mathcal{M}_2$ with q_2. From the answer it receives, it returns to query $\mathcal{M}_1$ again with query q_3. Eventually the adversary also sends a `privacy_loss` query and receives a response from the privacy-loss accumulator $\mathcal{G}$.

A valid $\mathcal{D}$ DP privacy-loss accumulator for interactive mechanisms is similarly defined as its non-interactive counterpart.

Definition 2.3.6 (Valid $\mathcal{D}$ DP privacy-loss accumulator for interactive mechanisms)**.** $\mathcal{G} : \mathcal{D}^* \to \mathcal{D}$ *is a valid $\mathcal{D}$ DP* concurrent IM-privacy-loss accumulator *if for every $d \in \mathcal{D}$, $\mathcal{G}(\cdot)$-OdomCon(IM) is a valid d-$\mathcal{D}$ DP odometer.*

Algorithm 6 $\mathcal{G}$-odometer for concurrent composition of interactive mechanisms ($\mathcal{G}$-*OdomCon*(IM)).

procedure $\mathcal{G}$-*OdomCon*(IM)(s, m):
  if $m = \lambda$ **then**
    $s' \leftarrow (s, [])$, where [] is an empty list
    return (s', λ)
  end if
  Parse $s = (x, [(\mathcal{M}_1, d_1, s_1), (\mathcal{M}_2, d_2, s_2), \ldots, (\mathcal{M}_k, d_k, s_k)])$
  Parse m
  if $m = (\mathcal{M}', d')$ **then**
    if $\mathcal{M}'$ is d'-$\mathcal{D}$ DP **then**
      $\mathcal{M}_{k+1} \leftarrow \mathcal{M}'$
      $d_{k+1} \leftarrow d'$
      $(s_{k+1}, m') \leftarrow \mathcal{M}_{k+1}(x, \lambda)$
      $s' \leftarrow (x, [(\mathcal{M}_1, d_1, s_1), \ldots, (\mathcal{M}_{k+1}, d_{k+1}, s_{k+1})])$
      $m' \leftarrow$ `yes`
    else
      $s' \leftarrow s, m' \leftarrow$ `"Divergence cannot be parsed"`
    end if
  else if $m = (j, q)$ where $j = 1, \ldots, k$ and q is a query to $\mathcal{M}_j$ **then**
    $(s'_j, m') \leftarrow \mathcal{M}_j(s_j, q)$
    $s' \leftarrow (x, [(\mathcal{M}_1, d_1, s_1), \ldots, (\mathcal{M}_j, d_j, s'_j), \ldots, (\mathcal{M}_k, d_k, s_k)])$
  else if $m =$ `privacy_loss` **then**
    $s' \leftarrow s, m' \leftarrow \mathcal{G}(d_1, \ldots, d_k)$
  else if m cannot be parsed correctly **then**
    $s' \leftarrow s, m' \leftarrow$ `invalid query`
  end if
  return (s', m')
end procedure

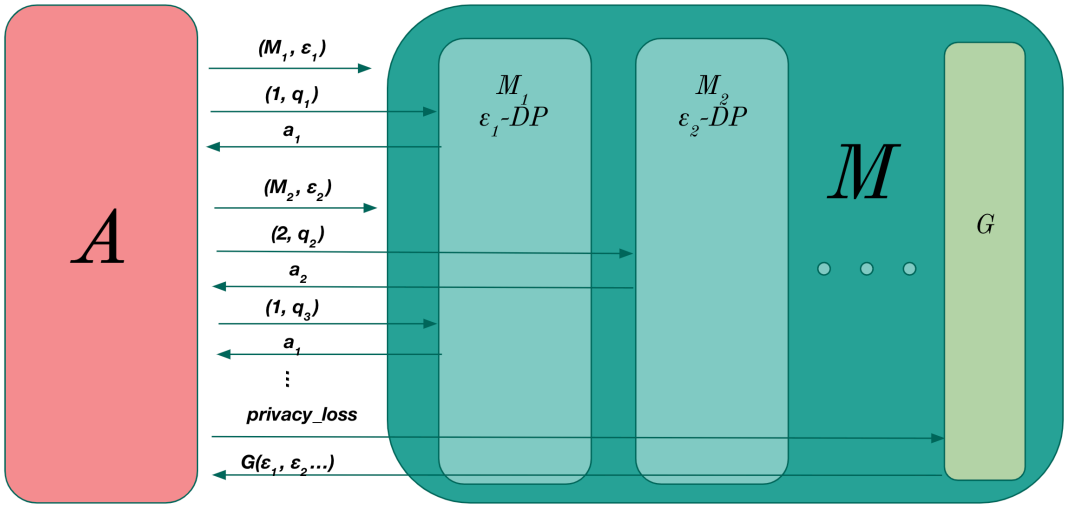


Figure 2.3: Schematic of a concurrent approximate-DP odometer.

2.4 I.M.-to-I.M. Postprocessing

Previous work [37,38] has defined interactive postprocessing of non-interactive mechanisms, as well as non-interactive postprocessing of non-interactive mechanisms, [11,42] both of which maintain privacy under the data-processing inequality. Defining interactive postprocessing of interactive mechanisms will be helpful in our proofs for concurrent, adaptive composition of I.M.s. If we already know the privacy properties of a particular I.M., and we know this I.M. can simulate another via interactive postprocessing, then we will also be able to make conclusions about the privacy of the simulated I.M. The task now is to define a primitive for such interactive postprocessing and show that the postprocessing itself preserves privacy.

A primitive that will be helpful to us in this setting is the *person-in-the-middle mechanism*, which can be thought of as an interlocutor between the mechanism and the adversary. Figure 2.4 is a representation of a person-in-the-middle mechanism.

Definition 2.4.1 (Person-in-the-middle mechanism)**.** *An* interactive postprocessing mechanism *(or*

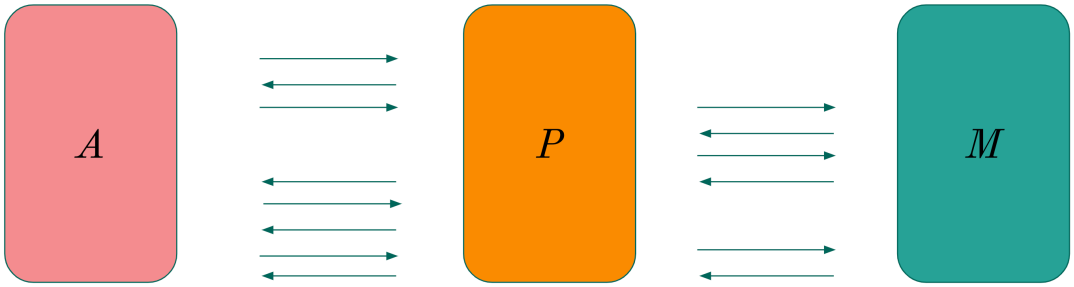


Figure 2.4: Person-in-the-middle mechanism. The *person-in-the-middle* $\mathcal{P}$ can communicate interactively with both adversary $\mathcal{A}$ and mechanism $\mathcal{M}$, as shown, with $\mathcal{A}$ choosing queries adaptively based on the messages it receives from $\mathcal{P}$. The message that $\mathcal{P}$ finally passes on can be thought of as the result of an interactive postprocessing.

'PIM mechanism' for 'person-in-the-middle') is a randomized mechanism $\mathcal{P} : \{0,1\}^* \times \{Q,A\} \times \{0,1\}^* \to \{0,1\}^* \times \{Q,A\} \times \{0,1\}^*$ *that takes its current state* $s \in \{0,1\}^*$*, a value* $v \in \{Q,A\}$ *to indicate whether it is receiving a query (from an analyst or adversary) or an answer (from the mechanism it is postprocessing), and a message* m*.* $\mathcal{P}$ *returns a new state* $s' \in \{0,1\}^*$*, a value* $v' \in \{Q,A\}$ *to indicate whether it is asking a query (of the mechanism it is postprocessing) or providing an answer (if it is answering a query) and message* m'*, denoted* $(s', v', m') = \mathcal{P}(s, v, m)$*.*

Intuitively, in an interaction, a PIM mechanism can engage adaptively with the adversary until it deems a (potentially-postprocessed) query appropriate to pass on to the mechanism. Upon receiving this query, the mechanism will produce an answer, and the PIM mechanism can undergo an adaptive interaction with the mechanism as well until it deems the message appropriate to pass back to the adversary. One can view v as an internal state that the PIM mechanism maintains.

Definition 2.4.2 (Postprocessed interactive mechanism)**.** *Let* $\mathcal{M}$ *be an interactive mechanism and* $\mathcal{P}$ *be an interactive postprocessing. Then the postprocessing of* $\mathcal{M}$ *by a PIM mechanism* $\mathcal{P}$*, denoted* $\mathcal{P} \circ \mathcal{M}$*, is the interactive mechanism defined in Algorithm 7.*

Essentially, though an adversary $\mathcal{A}$ only physically interacts with $\mathcal{P}$, the entire interaction $\mathcal{P} \circ \mathcal{M}$ can be viewed as a mechanism that $\mathcal{A}$ interacts with. This will be important in reduction proofs in

which we prove the privacy properties of one mechanism by transforming it into another whose privacy properties we already know.

Algorithm 7 Postprocessing of $\mathcal{M}$ by PIM mechanism $\mathcal{P}$, denoted $\mathcal{P} \circ \mathcal{M}$.

procedure $\mathcal{P} \circ \mathcal{M}(s, m)$:
  if $m = \lambda$ **then**
    $s^{\mathcal{M}} \leftarrow x, s^{\mathcal{P}} \leftarrow \lambda, m \leftarrow \lambda, v \leftarrow A$ ▷ initialize $\mathcal{M}$ and $\mathcal{P}$
  else
    Parse s as $s = (s^{\mathcal{M}}, s^{\mathcal{P}})$
    Let $v \leftarrow Q$
  end if
  while $v = Q$ **do** ▷ while $\mathcal{P}$ is equipped to interact with $\mathcal{M}$, conduct interaction
    $(s^{\mathcal{M}}, m) \leftarrow \mathcal{M}(s^{\mathcal{M}}, m)$
    $(s^{\mathcal{P}}, v, m) = \mathcal{P}(s^{\mathcal{P}}, A, m)$
  end while
  return $((s^{\mathcal{M}}, s^{\mathcal{P}}), m)$
end procedure

Theorem 2.4.3 (Privacy preserved under I.M.-to-I.M. postprocessing)**.** *If* $\mathcal{P} : \{0,1\}^* \times \{Q, A\} \times \{0,1\}^* \to \{0,1\}^* \times \{Q, A\} \times \{0,1\}^*$ *is an interactive postprocessing algorithm and mechanism* $\mathcal{M}$ *is d-*$\mathcal{D}$ *DP, then* $\mathcal{P} \circ \mathcal{M}$ *is also d-*$\mathcal{D}$ *DP.*

Proof. Let $\mathcal{A}$ be the deterministic adversary against $\mathcal{P} \circ \mathcal{M}$. By Definition 1.2.3, $\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{P} \circ \mathcal{M}(x)) = (r_0, m_1, r_2, m_3, r_4, \dots)$, where r_i are the random coins that $\mathcal{A}$ tosses and m_i are the messages received by $\mathcal{A}$.

Define an adversary $\mathcal{A}'$ that maintains the states of $\mathcal{A}$ and $\mathcal{P}$ as submachines, constructed as in Algorithm 8. Informally, Algorithm 8 says that when $\mathcal{A}'$ is to generate a query (determined by $v = Q$), it will run $\mathcal{A}$ with $\mathcal{P}$ until $\mathcal{P}$ decides the query is appropriate to pass on to $\mathcal{M}$ (determined by $v_{out} = Q$). When $\mathcal{A}'$ is to receive an answer (determined by $v = A$), it will run $\mathcal{P}$ to first check if $\mathcal{P}$ wants to interactively process it with $\mathcal{M}$ first. If so, the interaction between $\mathcal{P}$ and $\mathcal{M}$ would

proceed until $\mathcal{P}$ determines that the answer is in a state to be passed to $\mathcal{A}$, done by checking the output value v_{out} of $\mathcal{P}$ given the state, the value A, and the message as inputs at that round.

Algorithm 8 Adversary $\mathcal{A}'$ that maintains the states of $\mathcal{A}$ and $\mathcal{P}$ as submachines.

procedure $\mathcal{A}'(s, m)$:
  if $s = \lambda$ **then** ▷ need to initialize $\mathcal{A}'$
    $(s^{\mathcal{A}}, m_0) \leftarrow \mathcal{A}(\lambda, \lambda)$ ▷ initialize adversary submachine
    $(s^{\mathcal{P}}, v, m') \leftarrow \mathcal{P}(\lambda, A, m_0)$ ▷ initialize PIM submachine
    return $((s^{\mathcal{A}}, s^{\mathcal{P}}), m')$
  end if
  Parse $s = (s^{\mathcal{A}}, s^{\mathcal{P}})$
  $v_{out} \leftarrow A$ ▷ initialize v_{out}
  if $v = Q$ **then** ▷ $\mathcal{P}$ will provide a query to $\mathcal{M}$
    $j \leftarrow 1$ ▷ j indexes the internal communication $\mathcal{A} \leftrightarrow \mathcal{P}$
    while $v_{out} = A$ **do**
      Let $(s^{\mathcal{A}}_{j+1}, m_j) \leftarrow \mathcal{A}(s^{\mathcal{A}}_j, m)$
      Let $(s^{\mathcal{P}}_{j+1}, v_{out}, m'_j) \leftarrow \mathcal{P}(s^{\mathcal{P}}_{j+1}, v, m_j)$
      $j \leftarrow j + 1$
    end while
    $m' \leftarrow m'_j$
    $s' \leftarrow (s^{\mathcal{A}}_j, s^{\mathcal{P}}_j)$
    $v \leftarrow A$ ▷ prime $\mathcal{A}$ to receive an answer
  else if $v = A$ **then**
    $(s^{\mathcal{P}}, v_{out}, m') \leftarrow \mathcal{P}(s^{\mathcal{P}}, v, m)$
    $s' \leftarrow (s^{\mathcal{A}}, s^{\mathcal{P}})$
    if $v_{out} = A$ **then** ▷ $\mathcal{P}$ will provide an answer to $\mathcal{A}$
      $v \leftarrow Q$ ▷ prime $\mathcal{A}$ to ask a query
    end if
  end if
  return (s', m')
end procedure

Because $\mathcal{M}$ is d-$\mathcal{D}$ DP, the interaction between $\mathcal{A}'$ and $\mathcal{M}$ is d-$\mathcal{D}$ DP. Define a noninteractive postprocessing function g that transforms $\texttt{View}(\mathcal{A}' \leftrightarrow \mathcal{M}(x)) = (r'_0, m'_1, r'_2, m'_3, \dots)$ on any dataset x as in Algorithm 9. Informally, the new `View` object generated by Algorithm 9 makes visible

the submachine interaction between $\mathcal{A}$ and $\mathcal{P}$ in the operation of $\mathcal{A}'$, and snips the interaction between $\mathcal{A}'$ and $\mathcal{M}$ such that if there is an extended sequence between $\mathcal{M}$ and submachine $\mathcal{P}$ of $\mathcal{A}$, only the only the final answer from $\mathcal{M}$ before $\mathcal{P}$ and $\mathcal{A}$ begin interacting will be included. Since non-interactive postprocessing preserves privacy properties by Definition 1.5.1 (3), we know that $D(g(\texttt{View}(\mathcal{A}' \leftrightarrow \mathcal{M})(x))||g(\texttt{View}(\mathcal{A}' \leftrightarrow \mathcal{M})(x'))) \preceq D(\texttt{View}(\mathcal{A}' \leftrightarrow \mathcal{M})(x)||\texttt{View}(\mathcal{A}' \leftrightarrow \mathcal{M})(x')) \preceq d$.

The view of $\mathcal{A}$ against $\mathcal{P} \circ \mathcal{M}$ on any dataset x can be computed by postprocessing the view of $\mathcal{A}'$ against $\mathcal{M}$ on x using g; in other words, $\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{P} \circ \mathcal{M}) \equiv g(\texttt{View}(\mathcal{A}' \leftrightarrow \mathcal{M}))$ on any dataset x. Therefore, $D(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{P} \circ \mathcal{M})(x)||\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{P} \circ \mathcal{M})(x')) \preceq d$. This means $\mathcal{P} \circ \mathcal{M}$ is also d-$\mathcal{D}$ DP. □

This is a powerful result, as it shows that any state machine that operates as a PIM mechanism will, when composed with $\mathcal{M}$, maintain privacy properties.

A useful corollary follows from this theorem.

Corollary 2.4.4. *Suppose $\mathcal{N} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$ is an interactive mechanism such that for every pair of neighboring datasets x, x', every deterministic adversary $\mathcal{A}$, and every query $q \in \{0,1\}^*$, there exists an interactive mechanism $\mathcal{M}$ and a PIM mechanism $\mathcal{P}$ such that*

$$\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{N}(x,q)) \equiv \texttt{View}(\mathcal{A} \leftrightarrow \mathcal{P} \circ \mathcal{M}(x,q))$$
$$\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{N}(x',q)) \equiv \texttt{View}(\mathcal{A} \leftrightarrow \mathcal{P} \circ \mathcal{M}(x',q))$$

Proof. By Theorem 2.4.3, $\mathcal{P} \circ \mathcal{M}$ is d-$\mathcal{D}$ DP. Therefore, $D(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{N}(x,q))||\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{N}(x',q))) = D(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{P} \circ \mathcal{M}(x,q))||\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{P} \circ \mathcal{M}(x',q))) \preceq d$. Therefore, $\mathcal{N}$ is also d-$\mathcal{D}$ DP. □

Algorithm 9 Non-interactive postprocessing g that transforms $\texttt{View}(\mathcal{A}' \leftrightarrow \mathcal{M}(x))$ to $\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{P} \circ \mathcal{M}(x))$

procedure $g(\texttt{View}(\mathcal{A}' \leftrightarrow \mathcal{M}(x)))$
  $i \leftarrow 1$ ▷ i indexes input
  $j \leftarrow 0$ ▷ j indexes output
  Parse $\texttt{View}(\mathcal{A}' \leftrightarrow \mathcal{M}(x))$ as $(r'_0, m'_1, r'_2, m'_3, \dots)$
  $v, \leftarrow Q, v_{out} \leftarrow A$
  while m_i **do**
    if $v = Q$ **then** ▷ adding interaction $\mathcal{A} \leftrightarrow \mathcal{P}$
      Store $(s^{\mathcal{A}}, m^{\mathcal{A}}) \leftarrow (s_i^{\mathcal{A}}, m'_i)$ as temporary variable
      $(s^{\mathcal{P}}, v_{out}, m^{\mathcal{P}}) \leftarrow (s_i^{\mathcal{P}}, v, \lambda)$
      while $v_{out} \neq Q$ **do** ▷ simulate interaction of $\mathcal{A}$ and $\mathcal{P}$ on the message
        Let $(s^{\mathcal{A}}, m^{\mathcal{A}}) = \mathcal{A}(s^{\mathcal{A}}, m^{\mathcal{A}}; r_j^{\mathcal{A}})$
        Let $(s^{\mathcal{P}}, v_{out}, m^{\mathcal{P}}) = \mathcal{P}(s^{\mathcal{P}}, v, m^{\mathcal{A}}; r_j^{\mathcal{P}})$ ▷ using the randomness of $\mathcal{P}$ to get the new state
        $m_{j+1} \leftarrow m^{\mathcal{P}}$ ▷ record message $\mathcal{A}$ receives from $\mathcal{P}$
        $j \leftarrow j + 2$
      end while
      $v \leftarrow A$
    else if $v = A$ **then** ▷ abbreviating interaction $\mathcal{P} \leftrightarrow \mathcal{M}$
      $(s^{\mathcal{P}}, v_{out}, m) \leftarrow \mathcal{P}(s_i^{\mathcal{P}}, v, m'_i; r_j^{\mathcal{P}})$
      if $v_{out} = Q$ **then**
        $v \leftarrow A$
      else if $v_{out} = A$ **then** ▷ record the history
        $m_{j+1} \leftarrow m$
        $v \leftarrow Q$
        $j \leftarrow j + 2$
      end if
    end if
    $i \leftarrow i + 2$
  end while
  return $(r_0^{\mathcal{A}}, m_1, r_2^{\mathcal{A}}, m_3, \dots, r_j^{\mathcal{A}}, m_{j+1}, \dots)$
end procedure

2.5 Correspondence Between Odometers and Filters

A convenient property of odometers and filters is that any privacy-loss filter can be converted into a privacy-loss odometer, and vice versa. Given a valid privacy-loss accumulator $\mathcal{G}(\cdot)$, we can build a valid continuation rule $\mathcal{F}(\cdot; d) = \mathbb{I}(\mathcal{G}(\cdot) \preceq d)$. Similarly, for a valid continuation rule of the form $\mathcal{F}(\cdot; d) = \mathbb{I}(\mathcal{G}(\cdot) \preceq d)$, we can extract the function inside, $\mathcal{G}$, which will be a valid privacy-loss accumulator. We will use this correspondence throughout the proofs in Chapter 3 and 4.

Lemma 2.5.1. *1. A function $\mathcal{G} : \mathcal{D}^* \to \mathcal{D}$ is a valid NIM-privacy-loss accumulator if and only if $\mathcal{F} : \mathcal{D}^* \times \mathcal{D} \to \{0, 1\}$ constructed from $\mathcal{G}(\cdot)$ such that $\mathcal{F}(\cdot; d) = \mathbb{I}(\mathcal{G}(\cdot) \preceq d)$ is a valid $\mathcal{D}$-DP NIM-filter.*

2. A function $\mathcal{G} : \mathcal{D}^* \to \mathcal{D}$ is a valid concurrent IM-privacy-loss accumulator if and only if $\mathcal{F} : \mathcal{D}^* \times \mathcal{D} \to \{0, 1\}$ constructed from $\mathcal{G}(\cdot)$ such that $\mathcal{F}(\cdot; d) = \mathbb{I}(\mathcal{G}(\cdot) \preceq d)$ is a valid $\mathcal{D}$-DP concurrent IM-filter.

Proof of Lemma 2.5.1. We will prove Lemma 2.5.1 for the interactive case, because the non-interactive case follows similarly.

Fix a pair of datasets x, x'.

If. Suppose that $\mathcal{F}(\cdot; d) = \mathbb{I}(\mathcal{G}(\cdot) \preceq d)$ is a valid $\mathcal{D}$ DP concurrent IM-filter. Then by Definition 2.2.3, $\mathcal{F}(\cdot; d)$-*FiltCon*(IM) is a d-$\mathcal{D}$ DP interactive mechanism for every $d \in \mathcal{D}$. Thus, for an adversary $\mathcal{A}$ interacting with $\mathcal{G}$-*OdomCon*(IM), we can construct an interactive mechanism $\mathcal{M}$ that will behave exactly like $\mathcal{G}$-*OdomCon*(IM) as in Algorithm 6, but on every round of operation also checks if $\mathcal{G}(\cdot) \not\preceq d$, in which case $\mathcal{M}$ will maintain its previous state and output `halt` message. It is easy to see that $\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x))$ is exactly the same as $\texttt{Trunc}_d(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{G}\text{-}OdomCon(IM)(x)))$ for any database x.

We now prove that $\mathcal{M}$ is an interactive postprocessing of $\mathcal{F}(\cdot; d)$-*FiltCon*(IM). Define a postprocessing $\mathcal{P}$ as in Algorithm 10. This mechanism allows $\mathcal{M}$ to behave the same as $\mathcal{F}(\cdot; d)$-*FiltCon*(IM), just adding additional support for the deterministic `privacy_loss` queries, which do not change the state of the odometer and therefore do not affect divergence.

Therefore,

$$
\begin{aligned}
&D(\texttt{Trunc}_d(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{G}\text{-}OdomCon(IM)(x)))||\texttt{Trunc}_d(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{G}\text{-}OdomCon(IM)(x'))))\\
&= D(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x))||\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x')))\\
&= D(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{P} \circ \mathcal{F}\text{-}FiltCon(IM)(x)||\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{P} \circ \mathcal{F}\text{-}FiltCon(IM)(x')))\\
&\preceq D(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{F}\text{-}FiltCon(IM)(x))||\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{F}\text{-}FiltCon(IM)(x'))) \qquad (2.1)\\
&\preceq d, \qquad (2.2)
\end{aligned}
$$

where (2.1) follows from Theorem 2.4.3. By Definition 2.3.6, $\mathcal{G} : \mathcal{D}^* \to \mathcal{D}$ is a valid $\mathcal{D}$ DP privacy-loss accumulator.

Only if. Suppose that $\mathcal{G} : \mathcal{D}^* \to \mathcal{D}$ is a valid $\mathcal{D}$ DP privacy-loss accumulator. For every $d \in \mathcal{D}$, because the continuation rule is $\mathcal{F}(\cdot; d) = \mathbb{I}(\mathcal{G}(\cdot) \preceq d)$, $\mathcal{F}(\cdot; d)$-*FiltCon*(IM) will halt when $\mathcal{G}(\cdot) \not\preceq d$, which is the same condition for outputting `View` truncated at d as in Algorithm 5. We can use the same $\mathcal{M}$ as in the previous direction that is constructed such that, for an adversary $\mathcal{A}$ and database x, $\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x))$ is exactly the same as $\texttt{Trunc}_d(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{G}\text{-}OdomCon(IM)(x)))$.

We now define an interactive postprocessing $\mathcal{P}$ that transforms $\mathcal{M}$ into $\mathcal{F}(\cdot; d)$-*FiltCon*(IM) in Algorithm 11, which removes support for `privacy_loss` queries since $\mathcal{F}(\cdot; d)$-*FiltCon*(IM) would not support that. Other interaction is essentially preserved, since $\mathcal{M}$ and $\mathcal{F}(\cdot; d)$-*FiltCon*(IM) will both halt when $\mathcal{G}(\cdot) \not\preceq d$.

Again, because `privacy_loss` queries do not affect divergence, we have

$$
\begin{aligned}
&D(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{F}\text{-}\mathit{FiltCon}(\mathit{IM})(x)||\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{F}\text{-}\mathit{FiltCon}(\mathit{IM})(x')))\\
&= D(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{P} \circ \mathcal{M}(x)||\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{P} \circ \mathcal{M}(x')))\\
&\preceq D(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x))||\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x'))) \qquad (2.3)\\
&= D(\texttt{Trunc}_d(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{G}\text{-}\mathit{OdomCon}(\mathit{IM}))(x)||\texttt{Trunc}_d(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{G}\text{-}\mathit{OdomCon}(\mathit{IM}))(x')))\\
&\preceq d \qquad (2.4)
\end{aligned}
$$

where (2.3) follows from Theorem 2.4.3. Thus, $\mathcal{F}(\cdot; d)$-*FiltCon*(IM) is a d-$\mathcal{D}$ DP interactive mechanism. By Definition 2.2.3, since this is true for arbitrary d, $\mathcal{F}$ is a valid $\mathcal{D}$ DP filter. □

Algorithm 10 Interactive postprocessing $\mathcal{P}$ that transforms $\mathcal{F}(\cdot; d)$-*FiltCon*(IM) into $\mathcal{M}$

procedure $\mathcal{P}(s, v, m)$:
  $k \leftarrow$ number of mechanisms (started or attempting to start)
  if $m = \lambda$ **then** ▷ initialize $\mathcal{P}$
    $s' \leftarrow (\mathcal{G}(\cdot), [])$, $v \leftarrow A$, where [] is an empty list
    return (s', v, λ)
  end if
  if $v = Q$ **then** ▷ accepting queries from adversary
    if $m = (\mathcal{M}', d')$ **then**
      Parse s as $(\mathcal{G}(\cdot), [\mathcal{M}_1, \ldots, \mathcal{M}_k])$
      $s' \leftarrow (\mathcal{G}(\cdot), [\mathcal{M}_1, \ldots, \mathcal{M}_k, (\mathcal{M}_{k+1}, \texttt{pending})])$
    else if $m = (j, q)$ where $j = 1, \ldots, k$ and q is a query to $\mathcal{M}_j$ **then**
      $s' \leftarrow s$
    else if $m = \texttt{privacy_loss}$ **then**
      $m' \leftarrow \mathcal{G}(\cdot)$
      return (s', A, m')
    end if
    $m' \leftarrow m$
    $v' \leftarrow Q, v \leftarrow A$ ▷ pass query on to mechanism
  else if $v = A$ **then** ▷ accepting answers from filter
    if $m = \texttt{yes}$ **then** ▷ successfully started new mechanism
      Parse s as $(\mathcal{G}(\cdot), [\mathcal{M}_1, \ldots, (\mathcal{M}_k, \texttt{pending})])$
      $s' \leftarrow (\mathcal{G}(\cdot), [\mathcal{M}_1, \ldots, \mathcal{M}_k])$
    else if $m = \texttt{no}$ **then**
      Parse s as $(\mathcal{G}(\cdot), [\mathcal{M}_1, \ldots, (\mathcal{M}_k, \texttt{pending})])$
      $s' \leftarrow (\mathcal{G}(\cdot), [\mathcal{M}_1, \ldots, \mathcal{M}_{k-1}])$
    else
      $s' \leftarrow s$
    end if
    $m' \leftarrow m, v' \leftarrow A, v \leftarrow Q$ ▷ pass query on to adversary
  end if
  return (s', v', m')
end procedure

Algorithm 11 Interactive postprocessing $\mathcal{P}$ that transforms $\mathcal{M}$ into $\mathcal{F}(\cdot;d)$-*FiltCon*(IM)

procedure $\mathcal{P}(s, v, m)$:
  $k \leftarrow$ number of mechanisms (started or attempting to start)
  if $m = \lambda$ **then** ▷ initialize $\mathcal{P}$
    $s' \leftarrow (\mathcal{G}(\cdot), [])$, $v \leftarrow A$, where [] is an empty list
    return (s', v, λ)
  end if
  if $v = Q$ **then** ▷ accepting queries from adversary
    if $m = (\mathcal{M}', d')$ **then**
      Parse s as $(\mathcal{G}(\cdot), [\mathcal{M}_1, \ldots, \mathcal{M}_k])$
      $s' \leftarrow (\mathcal{G}(\cdot), [\mathcal{M}_1, \ldots, \mathcal{M}_k, (\mathcal{M}_{k+1}, \texttt{pending})])$
    else if $m = (j, q)$ where $j = 1, \ldots, k$ and q is a query to $\mathcal{M}_j$ **then**
      $s' \leftarrow s$
    else if $m =$ `privacy_loss` **then** ▷ remove support for privacy loss queries
      $m' \leftarrow$ `invalid query`
      return (s', A, m')
    end if
    $m' \leftarrow m$
    $v' \leftarrow Q, v \leftarrow A$ ▷ pass query on to mechanism
  else if $v = A$ **then** ▷ accepting answers from filter
    if $m =$ `yes` **then** ▷ successfully started new mechanism
      Parse s as $(\mathcal{G}(\cdot), [\mathcal{M}_1, \ldots, (\mathcal{M}_k, \texttt{pending})])$
      $s' \leftarrow (\mathcal{G}(\cdot), [\mathcal{M}_1, \ldots, \mathcal{M}_k])$
    else if $m =$ `no` **then**
      Parse s as $(\mathcal{G}(\cdot), [\mathcal{M}_1, \ldots, (\mathcal{M}_k, \texttt{pending})])$
      $s' \leftarrow (\mathcal{G}(\cdot), [\mathcal{M}_1, \ldots, \mathcal{M}_{k-1}])$
    else
      $s' \leftarrow s$
    end if
    $m' \leftarrow m, v' \leftarrow A, v \leftarrow Q$ ▷ pass query on to adversary
  end if
  return (s', v', m')
end procedure

3

Concurrent Filters and Odometers for *f*-DP

3.1 *f*-Differential Privacy

Differential privacy attempts to capture privacy loss by taking the statistical divergence of the output of a mechanism on adjacent datasets. *f*-DP [10] casts this formulation into a hypothesis testing setting. The adversary, upon receiving the output of some mechanism $\mathcal{M}$ on a dataset, considers the

following problem:

$$H_0 : \text{the dataset is } x \qquad\qquad H_1 : \text{the dataset is } x'$$

Let Y and Y' denote the output of $\mathcal{M}$ on two adjacent datasets, i.e. $Y = \mathcal{M}(x)$ and $Y' = \mathcal{M}(x')$. Let φ be a rejection rule mapping the output of a random variable to a binary decision to reject or fail to reject. The *type I error*, denoted $\alpha_\varphi = \mathrm{E}[\varphi(Y)]$, is the probability of rejecting H_0 when H_0 is true. The *type II error*, denoted $\beta_\varphi = 1 - \mathrm{E}[\varphi(Y')]$, is the probability of failing to reject H_0 when H_1 is true. For a given confidence level, the trade-off function describes the greatest lower bound of the type II error for φ. Essentially, the trade-off function characterizes the optimal boundary between the minimal achievable type I and type II errors.

Definition 3.1.1 (Trade-off function [10]). *Let Y and Y' be two probability distributions on the same space. Then the* trade-off function $T(Y, Y') : [0,1] \to [0,1]$ *is given by*

$$T(Y, Y')(\alpha) = \inf_\varphi \{\beta_\varphi : \alpha_\varphi \leq \alpha\}.$$

The necessary and sufficient conditions of trade-off functions $f : [0,1] \to [0,1]$ are that each is convex, continuous, nonincreasing, and $f(\alpha) \leq 1 - f(\alpha)$ on $[0,1]$. [10]

Definition 3.1.2 (f-DP [10]). *Let $f : [0,1] \to [0,1]$ be a trade-off function. A mechanism $\mathcal{M}$ is* f-differentially private *if for every pair of neighboring datasets x, x',*

$$T(\mathcal{M}(x), \mathcal{M}(x')) \geq f.$$

Essentially, for any confidence level, the type II error is always bounded below by f. Hence, the larger the function f on $[0,1]$, the more difficult it is to distinguish between $\mathcal{M}(x)$ and $\mathcal{M}(x')$, and

hence the stronger the privacy guarantee.

f-DP is powerful in that it is a *complete* characterization of privacy, as it is a function definition rather than a pointwise definition. Hence, there is no risk of summarizing privacy loss too early based on insufficient statistical information.

In our generalized DP framework given in Definition 1.5.1, let the poset $(\mathcal{D}, \preceq)$ consist of the set $\mathcal{D}$ of all trade-off functions f. For $f_1, f_2 \in \mathcal{D}$, the partial ordering is $f_1 \preceq f_2$ if $f_1(\alpha) \geq f_2(\alpha)$ for all $\alpha \in [0,1]$. Let the mapping D be the trade-off function T as defined in Definition 3.1.1.

Example 3.1.3 ((ε, δ)-DP as a specific case of f-DP[40,22,10]). *Let* $f_{\varepsilon,\delta} = \max\{0, 1-\delta-\exp(\varepsilon)\alpha, \exp(-\varepsilon)(1-\delta-\alpha)\}$. *A mechanism is* $f_{\varepsilon,\delta}$*-DP iff it is* (ε, δ)*-DP.*

Proof. Fix a pair of neighboring datasets x, x', and let $\mathcal{M}$ be some (ε, δ)-DP mechanism. Define hypothesis H_0 to be that the dataset is x and H_1 to be that the dataset is x'. Let φ be the rejection rule that allows probabilistic decisions based on a partition of sets $\{\mathcal{S}_i\}, \mathcal{S}_i \subseteq \{0,1\}^*$ of the output space of an $f_{\varepsilon,\delta}$-DP mechanism $\mathcal{M}$. For a set $\mathcal{S}_i$, φ will reject the null hypothesis with probability p_i. Hence, for any such partition, the type I error $\alpha = \mathrm{E}[\varphi(\mathcal{M}(x))] = \sum_i p_i \Pr[\mathcal{M}(x) \in \mathcal{S}_i]$, and the type II error $\beta = 1 - \mathrm{E}[\varphi(\mathcal{M}(x'))] = \sum_i (1-p_i) \Pr[\mathcal{M}(x') \in \mathcal{S}_i]$.

An (ε, δ)-DP mechanism is $f_{\varepsilon,\delta}$-DP. Consider an arbitrary set partition $\{\mathcal{S}_i\}$. By the definition of (ε, δ)-DP as given in Definition 1.1.1, $\Pr[\mathcal{M}(x) \in \mathcal{S}_i] \leq \exp(\varepsilon) \Pr[\mathcal{M}(x') \in \mathcal{S}_i] + \delta$ for any $\mathcal{S}_i$. Then, $(1-p_i)\Pr[\mathcal{M}(x) \in \mathcal{S}_i] \leq (1-p_i)(\exp(\varepsilon)\Pr[\mathcal{M}(x') \in \mathcal{S}_i] + \delta)$, so $\sum_i (1-p_i)\Pr[\mathcal{M}(x) \in \mathcal{S}_i] \leq \exp(\varepsilon)\sum_i (1-p_i)\Pr[\mathcal{M}(x') \in \mathcal{S}_i] + \delta$, which rearranges to $1 - \alpha \leq \exp(\varepsilon)\beta + \delta$. This rearranges to $\beta \geq \exp(-\varepsilon)(1-\delta-\alpha)$. The (ε, δ)-DP guarantee also ensures that $\Pr[\mathcal{M}(x') \in \mathcal{S}_i] \leq \exp(\varepsilon)\Pr[\mathcal{M}(x) \in \mathcal{S}_i] + \delta$, so $1 - \beta \leq \exp(\varepsilon)\alpha + \delta \implies \beta \geq 1 - \delta - \exp(\varepsilon)\alpha$.

The tradeoff function should ensure that $\beta \geq \alpha$ at any level of α, so given the relationships we derived above, we know that $\beta \geq \max\{0, 1-\delta-\exp(\varepsilon)\alpha, \exp(-\varepsilon)(1-\delta-\alpha)\}$. Hence, an (ε, δ)-DP mechanism $\mathcal{M}$ is $f_{\varepsilon,\delta}$-DP, where $f_{\varepsilon,\delta}$ is defined above.

An $f_{\varepsilon,\delta}$-DP mechanism is (ε,δ)-DP. Note that $f_{\varepsilon,\delta}$ is a linear piecewise function symmetric with respect to the line $f_{\varepsilon,\delta}(\alpha) = \alpha$, with slope $-\exp(\varepsilon)$ when $f_{\varepsilon,\delta}(\alpha) > \alpha$ and $-\exp(-\varepsilon)$ when $f_{\varepsilon,\delta}(\alpha) < \alpha$. We will examine every piece. Knowing that $f_{\varepsilon,\delta}$ is the tradeoff function means $\beta \geq 1-\delta-\exp(\varepsilon)\alpha$ and $\beta \geq \exp(-\varepsilon)(1-\delta-\alpha)$. For the first piece,

$$\begin{aligned}
&\Pr[\varphi(\mathcal{M}(x')) = 0] \geq 1-\delta-\exp(\varepsilon)\Pr[\varphi(\mathcal{M}(x)) = 1] \\
\implies &\sum_i (1-p_i)\Pr[\mathcal{M}(x') \in \mathcal{S}_i] \geq 1-\delta-\exp(\varepsilon)\sum_i p_i \Pr[\mathcal{M}(x) \in \mathcal{S}_i] \\
\implies &1-\sum_i p_i \Pr[\mathcal{M}(x') \in \mathcal{S}_i] \geq 1-\delta-\exp(\varepsilon)\sum_i p_i \Pr[\mathcal{M}(x) \in \mathcal{S}_i] \\
\implies &\sum_i p_i \Pr[\mathcal{M}(x') \in \mathcal{S}_i] \leq \exp(\varepsilon)\sum_i p_i \Pr[\mathcal{M}(x) \in \mathcal{S}_i] + \delta
\end{aligned}$$

Because this holds for any arbitrary partition $\{\mathcal{S}_i\}$, and any arbitrary set $\mathcal{S} \subseteq \{0,1\}^*$ in $\mathcal{M}$'s output space can be constructed from some such partition (consider the partition $\mathcal{S}_1 = \mathcal{S}$ and $\mathcal{S}_2 = \overline{\mathcal{S}}$, with $p_1 = 1$ and $p_0 = 0$), we have that $\Pr[\mathcal{M}(x') \in \mathcal{S}] \leq \exp(\varepsilon)\Pr[\mathcal{M}(x) \in \mathcal{S}] + \delta$ for any set $\mathcal{S}$. For the second piece,

$$\begin{aligned}
&\Pr[\varphi(\mathcal{M}(x')) = 0] \geq \exp(-\varepsilon)(1-\delta-\Pr[\varphi(\mathcal{M}(x)) = 1]) \\
\implies &1-\sum_i p_i \Pr[\mathcal{M}(x) \in \mathcal{S}_i] \leq \exp(\varepsilon)(1-\sum_i p_i \Pr[\mathcal{M}(x') \in \mathcal{S}_i]) + \delta
\end{aligned}$$

Taking our partition for any arbitrary set $\mathcal{S}$ again, we have $1-\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(\varepsilon)(1-\Pr[\mathcal{M}(x') \in \mathcal{S}]) + \delta \implies \Pr[\varphi(\mathcal{M}(x)) = 0] \leq \exp(\varepsilon)\Pr[\varphi(\mathcal{M}(x')) = 0] + \delta \implies \Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(\varepsilon)\Pr[\mathcal{M}(x') \in \mathcal{S}] + \delta$ for any $\mathcal{S}$.

Hence, the definition of (ε,δ)-DP is satisfied, which implies an $f_{\varepsilon,\delta}$ mechanism is (ε,δ)-DP as well. □

A nice property of f-DP is that any interactive mechanism can be simulated by a non-interactive

mechanism, under an interactive postprocessing.

Theorem 3.1.4 (Reduction of f-DP into non-interactive mechanism[38])**.** *For every f in a set of trade-off functions $\mathcal{F}$, every pair of adjacent datasets x, x', and every interactive f-DP mechanism $\mathcal{M}$ with finite communication complexity, there exists a pair of random variables Y, Y' with support $[0,1]$ and a randomized interactive mechanism $\mathcal{P}$ such that $D(Y, Y') \preceq f$.*

For every deterministic adversary $\mathcal{A}$, we have

$$\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x)) \equiv \texttt{View}(\mathcal{A} \leftrightarrow \mathcal{P}(Y))$$
$$\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x')) \equiv \texttt{View}(\mathcal{A} \leftrightarrow \mathcal{P}(Y'))$$

Essentially, Y, Y' can be seen as the outputs of some non-interactive mechanism $\mathcal{N}$ on datasets x, x' respectively, i.e. $Y := \mathcal{N}(x)$ and $Y' := \mathcal{N}(x')$. Theorem 3.1.4 says that even if we give Y and Y' to an adversary – in this context, a hypothesis tester – it cannot distinguish them better than it can distinguish $\mathcal{M}$ on x and x'. Because $\mathcal{M}$ is a randomized, stateful mechanism, an adversary interacting with $\mathcal{M}$ may practically never know how $\mathcal{M}$ would answer *all* queries over the course of computation because the answer to each of $\mathcal{M}$'s queries depends on the previous interaction. This contrasts with ordinary hypothesis testing, which is a non-interactive task in which the adversary collects a full sample from the distribution and determines whether or not to reject H_0. However, with this theorem, even if we were to give Y and Y' to $\mathcal{A}$ directly, revealing how $\mathcal{M}$ would behave in response to *all* queries on x and x', $\mathcal{A}$ will still not be able to tell Y, Y' apart better than it can tell apart $\mathcal{M}(x), \mathcal{M}(x')$. The caveat is that this simulation only works for a *specific* interaction with $\mathcal{M}$ on x and x', with no rewinding to explore other paths in the interaction tree.

By the data-processing inequality, interactive postprocessing of non-interactive mechanisms would preserve privacy properties. Hence, it suffices to consider non-interactive mechanisms when proving privacy properties of interactive ones.

In this chapter, we will show that a concurrent filter for interactive mechanisms can be constructed without any privacy loss from a concurrent filter for non-interactive mechanisms, whose privacy properties have been shown in previous literature.[38] We will demonstrate an application of our theorem by presenting an example of an (ε, δ) concurrent filter and concurrent odometer.

3.2 Concurrent Filter & Odometer for f-DP

Using Theorem 3.1.4, we derive a concurrent f-DP filter from an f-DP filter for non-interactive mechanisms.

Theorem 3.2.1 (Concurrent f-DP filter)**.** *Suppose that $\mathcal{F} : \mathcal{D}^* \to \{1, 0\}$ is a valid f-DP continuation rule for non-interactive mechanisms. Then $\mathcal{F}$ is a valid f-DP continuation rule for interactive mechanisms with finite communication.*

Proof. Fix a pair of adjacent datasets x, x'. For any such pair, we can define the I.M.-to-I.M. postprocessing $\mathcal{P}$ that simluates $\mathcal{F}\text{-}FiltCon(IM)$ using $\mathcal{F}\text{-}Filt(NIM)$. For simplicity, we will assume that $\mathcal{A}$ is deterministic, as Lemma 1.5.5 allows us to do. By the postprocessing property of f-DP IMs given in Theorem 3.1.4, we know that an f_j-DP mechanism $\mathcal{M}_i$ can be simulated by an interactive postprocessing $\mathcal{P}_j$ of a non-interactive f_j-DP mechanism $\mathcal{N}_j$.

$\mathcal{P}$ is constructed as follows. $\mathcal{P}$ depends on whether it is primed to receive a query $v = Q$ from the adversary or an answer $v = A$ from the mechanism, to pass to the other party. If the adversary asks to start a new interactive mechanism, $\mathcal{P}$ will make note of the corresponding non-interactive mechanism and the non-interactive-to-interactive postprocessing and pass the request on $\mathcal{F}\text{-}FiltSeq(NIM)$. Otherwise, if the adversary queries an existing interactive mechanism $\mathcal{M}_j$, $\mathcal{P}$ will use the corresponding interactive postprocessing $\mathcal{P}_j$ to answer the query instead of passing the message forward to $\mathcal{M}$. The only time $\mathcal{P}$ will interact with $\mathcal{M}$ is to start a new mechanism and to pass back the con-

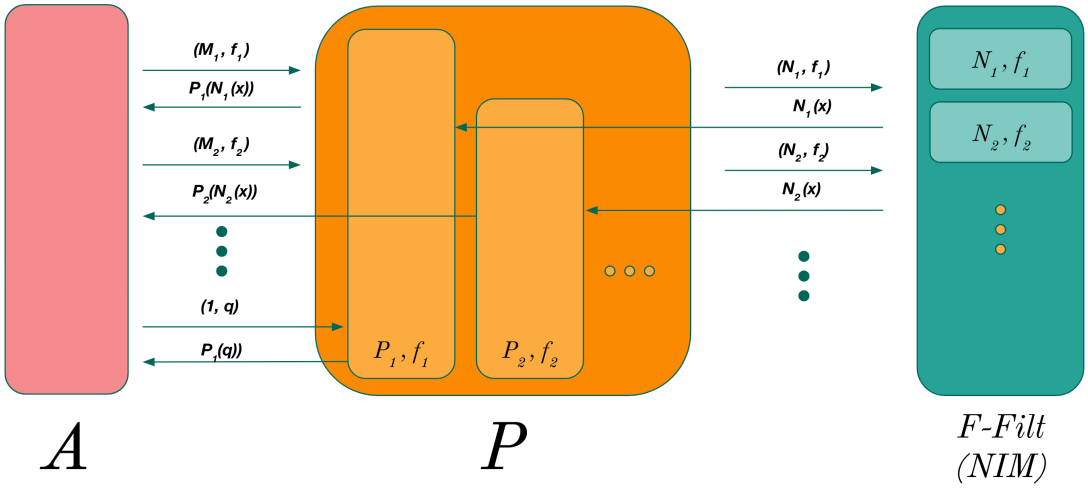


Figure 3.1: Schematic of IM-to-IM postprocessing $\mathcal{P}$ for Theorem 3.2.1. $\mathcal{P}$ will only send requests to start new mechanisms to $\mathcal{F}$-*Filt*(NIM); for queries to existing mechanisms, $\mathcal{P}$ will use the corresponding interactive NIM-to-IM postprocessing to answer the query.

firmation that a new mechanism has begun to $\mathcal{A}$. The algorithmic pseudocode of $\mathcal{P}$ is in Algorithm 12, and the schema is depicted in Figure 3.1.

Note that this $\mathcal{P}$ only interacts with $\mathcal{F}$-*FiltSeq*(NIM) in starting a new mechanism. For the rest of the interactive queries, the algorithm directly uses the interactive postprocessing of the corresponding non-interactive mechanism in question to answer the queries. By Theorem 3.1.4, we know that for every deterministic adversary $\mathcal{A}$, there exists an $\mathcal{F}$-*Filt*(NIM) that is f-DP and an interactive postprocessing $\mathcal{P}$ such that $\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{P} \circ \mathcal{F}\text{-}Filt(NIM)) \equiv \texttt{View}(\mathcal{A} \leftrightarrow \mathcal{F}\text{-}FiltCon(IM))$ on any database x. Because $\mathcal{F}$-*Filt*(NIM) is f-DP, by Theorem 2.4.3, $\mathcal{F}$-*FiltCon*(IM) is f-DP. □

An analogue of Theorem 3.2.1 holds for (ε, δ)-DP.

Corollary 3.2.2 (Concurrent (ε, δ) filter)**.** *Every filter for non-interactive (ε, δ)-DP mechanisms is also a concurrent filter of interactive (ε, δ)-DP mechanisms.*

Concurrent f-DP odometers can be defined similarly from f-DP odometers for non-interactive mechanisms.

Algorithm 12 Interactive postprocessing $\mathcal{P}$ that transforms f-DP $\mathcal{F}$-*Filt*(*NIM*) into $\mathcal{F}$-*FiltCon*(*IM*).

procedure $\mathcal{P}(s, v, m; r)$:
  $k \leftarrow$ number of mechanisms (started or attempting to start)
  if $m = \lambda$ **then** ▷ initialize $\mathcal{P}$
    $s' \leftarrow \lambda, v \leftarrow \mathcal{A}$
    return (s', v, λ)
  end if
  if $v = Q$ **then** ▷ $\mathcal{P}$ accepting queries
    if $m = (\mathcal{M}', f')$ **then** ▷ $\mathcal{A}$ starts a new mechanism
      Let $\mathcal{P}_{k+1}$ be the interactive postprocessing for non-interactive mechanism $\mathcal{N}_{k+1}$ so that $\mathcal{P}_{k+1}(\mathcal{N}_{k+1}) \equiv \mathcal{M}_{k+1}$
      Parse $s^{\mathcal{P}}$ as $((\mathcal{P}_1, s^{\mathcal{P}_1}, f_1), \ldots, (\mathcal{P}_k, s^{\mathcal{P}_k}, f_k))$
      $s^{\mathcal{P}'} \leftarrow ((\mathcal{P}_1, s^{\mathcal{P}_1}, f_1), \ldots, (\mathcal{P}_{k+1}, \texttt{pending}, f_{k+1}))$
      $m' \leftarrow (\mathcal{N}_{k+1}, f_{k+1})$
    else if $m = (j, q)$ where $j = 1, \ldots, k$ and q is a query to $\mathcal{M}_j$ **then** ▷ $\mathcal{A}$ queries an old mechanism
      Parse $s^{\mathcal{P}}$ as $((\mathcal{P}_1, s^{\mathcal{P}_1}, f_1), \ldots, (\mathcal{P}_k, s^{\mathcal{P}_k}, f_k))$
      $(s^{\mathcal{P}'_j}, m') \leftarrow \mathcal{P}_j(s^{\mathcal{P}_j}, q)$
      $s^{\mathcal{P}'} \leftarrow ((\mathcal{P}_1, s^{\mathcal{P}_1}, f_1), \ldots, (\mathcal{P}_j, s^{\mathcal{P}'_j}, f_j), \ldots (\mathcal{P}_k, s^{\mathcal{P}_k}, f_k))$
    end if
    return $(s^{\mathcal{P}'}, \mathcal{A}, m')$
  else if $v = \mathcal{A}$ **then** ▷ $\mathcal{P}$ accepting answers
    if $m = \mathsf{no}$ **then** ▷ did not successfully start a new mechanism
      $s^{\mathcal{P}'} \leftarrow ((\mathcal{P}_1, s^{\mathcal{P}_1}, f_1), \ldots, (\mathcal{P}_k, s^{\mathcal{P}_k}, f_k))$
      $m' \leftarrow m$
    else ▷ non-interactive output to starting a new mechanism
      Parse $s^{\mathcal{P}}$ as $((\mathcal{P}_1, s^{\mathcal{P}_1}, f_1), \ldots, (\mathcal{P}_k, \texttt{pending}, f_k))$
      $(s^{\mathcal{P}_k}, m') \leftarrow \mathcal{P}(\lambda, \mathcal{A}, m)$ ▷ set up new interactive postprocessor's state
      $s^{\mathcal{P}'} \leftarrow ((\mathcal{P}_1, s^{\mathcal{P}_1}, f_1), \ldots, (\mathcal{P}_k, s^{\mathcal{P}_k}, f_k))$
      $v \leftarrow Q$ ▷ update v to be accepting queries
    end if
    return $(s^{\mathcal{P}'}, \mathcal{A}, m')$
  end if
end procedure

Theorem 3.2.3 (Concurrent f-DP odometer)**.** *Suppose that $\mathcal{G} : \mathcal{D}^* \to \mathcal{D}$ is a valid f-DP privacy-loss accumulator for non-interactive mechanisms. Then $\mathcal{G}$ is a valid f-DP privacy-loss accumulator for interactive mechanisms with finite communication.*

Proof. By Lemma 2.5.1, if $\mathcal{G}$ is a valid f-DP privacy-loss accumulator, then $\mathcal{F}(\cdot; f) = I(\mathcal{G}(\cdot) \preceq f)$ is a valid f-DP filter for non-interactive mechanisms. By Theorem 3.2.1, if $\mathcal{F}$ is an f-DP filter for non-interactive mechanisms, then $\mathcal{F}$ will be an f-DP concurrent filter for interactive mechanisms. Applying Lemma 2.5.1 again, $\mathcal{G}$ will be a valid f-DP privacy-loss accumulator for interactive mechanisms, and $\mathcal{G}\text{-}OdomCon(IM)$ is an f-DP interactive mechanism. □

As a note, there are currently no known theorems for fully-adaptive sequential composition of f-DP. Our results for concurrent filters and odometers, therefore, also fill in this research gap.

3.3 Sample f-DP Filter & Odometer

Now we give an example of how to construct a valid (ε, δ)-DP privacy filter and odometer.

Theorem 3.3.1 ((ε, δ)-DP privacy filter[41])**.** *For every $\delta' > 0$,*

$$\begin{aligned}&\mathcal{F}((\varepsilon_1, \delta_1), \ldots, (\varepsilon_k, \delta_k); (\varepsilon, \delta)) \\ &= \mathbb{I}\left(\varepsilon \leq \sqrt{2 \ln\left(\frac{1}{\delta'}\right) \sum_{m \leq k} \varepsilon_m^2} + \frac{1}{2} \sum_{m \leq k} \varepsilon_m^2\right) \cdot \mathbb{I}\left(\delta' + \sum_{m \leq k} \delta_k \leq \delta\right)\end{aligned}$$

is a valid approx-DP continuation rule for non-interactive mechanisms and interactive mechanisms.

Proof. This filter is translated from the following theorem from Whitehouse et al.[41], who define an (ε, δ)-DP privacy filter in their notation as follows:

Theorem 3.3.2 ((ε, δ)-DP privacy filter[41])**.** *Suppose $\mathcal{M}_1, \mathcal{M}_2, \ldots$ is a sequence of mechanisms such that, for any $k \geq 1$, $\mathcal{M}_k$ is $(\varepsilon_k, \delta_k)$-DP conditioned on $\mathcal{M}_1, \ldots, \mathcal{M}_{k-1}$. Let $\varepsilon > 0$ and*

$\delta = \delta' + \delta''$ *be a target approximation parameter such that* $\delta' > 0, \delta'' \geq 0$. *Define* $N := N((\varepsilon_1, \delta_1), (\varepsilon_2, \delta_2), \dots) := \inf\{k \in \mathbb{N} : \varepsilon < \sqrt{2 \ln\left(\frac{1}{\delta'}\right) \sum_{m \leq k+1} \varepsilon_m^2} + \frac{1}{2} \sum_{m \leq k+1} \varepsilon_m^2$ *or* $\delta'' < \sum_{m \leq k+1} \delta_m\}$. *Then the composition of these mechanisms Comp*$(\mathcal{M}_1, \dots, \mathcal{M}_{N(\cdot)})$ *is* (ε, δ)*-DP.*

$\mathcal{M}_k$ being $(\varepsilon_k, \delta_k)$-DP conditioned on the previous mechanisms means the privacy-loss parameters are chosen adaptively. This is analogous to our setting for filters as defined in Algorithm 2, in which the adversary can send in a query that decomposes into a mechanism and distance, a signal to the filter to start a new mechanism. Since the query could depend on the history of interaction, the new mechanism's parameters are adaptively chosen.

We define $\mathcal{F} : \mathcal{D}^* \to \{1, 0\}$ as

$$\begin{aligned} &\mathcal{F}((\varepsilon_1, \delta_1), \dots, (\varepsilon_k, \delta_k); (\varepsilon, \delta)) \\ &= \mathbb{I}\left(\sqrt{2 \ln\left(\frac{1}{\delta'}\right) \sum_{m \leq k} \varepsilon_m^2} + \frac{1}{2} \sum_{m \leq k} \varepsilon_m^2 \leq \varepsilon\right) \cdot \mathbb{I}\left(\delta' + \sum_{m \leq k} \delta_m \leq \delta\right), \end{aligned}$$

meaning that a mechanism equipped with $\mathcal{F}$ will continue until for some k, $\sqrt{2 \ln\left(\frac{1}{\delta'}\right) \sum_{m \leq k+1} \varepsilon_m^2} + \frac{1}{2} \sum_{m \leq k+1} \varepsilon_m^2 > \varepsilon$ or $\sum_{m \leq k+1} \delta_m > \delta''$. This is equivalent to Whitehouse et al's[41] notion of $N(\cdot)$. Both the condition on ε and the condition on δ'' are increasing in the number of mechanisms being composed. The difference is that in Whitehouse et al.'s[41] setting, the number of mechanisms being composed $N(\cdot)$ is the smallest number for adding another mechanism will cause $\sqrt{2 \ln\left(\frac{1}{\delta'}\right) \sum_{m \leq N(\cdot)+1} \varepsilon_m^2} + \frac{1}{2} \sum_{m \leq N(\cdot)+1} \varepsilon_m^2$ to exceed ε, or in other words, $N(\cdot)$ is the largest number for which $\sqrt{2 \ln\left(\frac{1}{\delta'}\right) \sum_{m \leq N(\cdot)} \varepsilon_m^2} + \frac{1}{2} \sum_{m \leq N(\cdot)} \varepsilon_m^2 \leq \varepsilon$. Since we define our filter in terms of a bound on the privacy parameter instead of a bound on the number of mechanisms, this is equivalent to checking that for k mechanisms being composed, $\sqrt{2 \ln\left(\frac{1}{\delta'}\right) \sum_{m \leq k} \varepsilon_m^2} + \frac{1}{2} \sum_{m \leq k} \varepsilon_m^2 \leq \varepsilon$. The same holds for the $\sum_{m \leq k} \delta_m \leq \delta''$ term. Since δ' is already set, we fold the $\sum_{m \leq k} \delta_m \leq \delta''$ requirement into the $\delta' + \delta'' \leq \delta$ requirement to get the condition $\delta' + \sum_{m \leq k} \delta_m \leq$

δ. Taking the lower bound of the k that satisfies either $\sqrt{2\ln\left(\frac{1}{\delta'}\right)\sum_{m\le k}\varepsilon_m^2} + \frac{1}{2}\sum_{m\le k}\varepsilon_m^2 \le \varepsilon$ or $\delta' + \sum_{m\le k}\delta_m \le \delta$ is equivalent to finding the k that satisfies both conditions.

With fully adaptive composition of non-interactive mechanisms, the mechanism will only halt if starting a new mechanism would exceed its privacy-loss budget. Since $N(\cdot)$ mechanisms composed are (ε, δ)-DP in Whitehouse et al.[41]'s setting, we know that a mechanism equipped with $\mathcal{F}$ that has not yet halted will be (ε, δ)-DP as well.

Then by Definition 2.2.3, $\mathcal{F}$-*Filt*(NIM) is an (ε, δ)-DP interactive mechanism, meaning that $\mathcal{F}$ is a valid (ε, δ)-DP for the sequence of $(\varepsilon_i, \delta_i)$-DP non-interactive mechanisms as defined above. □

Theorem 3.3.3 ((ε, δ)-DP privacy odometer)**.** *Let $\delta = \delta' + \delta''$ be a target approximation parameter such that $\delta' > 0$, $\delta'' \ge 0$. Then*

$$\mathcal{G}((\varepsilon_1, \delta_1), \ldots, (\varepsilon_k, \delta_k)) = \begin{cases} \left(\sqrt{2\ln\left(\frac{1}{\delta'}\right)\sum_{m\le k}\varepsilon_m^2} + \frac{1}{2}\sum_{m\le k}\varepsilon_m^2, \delta\right) & \text{if } \delta' + \sum_{m\le k}\delta_m \le \delta \\ (\infty, \infty) & \text{otherwise} \end{cases}$$

is a valid approximate-DP privacy-loss accumulator for non-interactive and interactive mechanisms, where $\varepsilon = \sqrt{2\ln\left(\frac{1}{\delta'}\right)\sum_{m\le k}\varepsilon_m^2} + \frac{1}{2}\sum_{m\le k}\varepsilon_m^2$.

To prove Theorem 3.3.3, a concurrent privacy odometer, we make use of Lemma 2.5.1.

Proof. Note that

$$\begin{aligned} &\mathcal{F}((\varepsilon_1, \delta_1), \ldots, (\varepsilon_k, \delta_k); (\varepsilon, \delta)) \\ &= \mathbb{I}(\mathcal{G}((\varepsilon_1, \delta_1), \ldots, (\varepsilon_k, \delta_k) \le (\varepsilon, \delta)). \end{aligned}$$

By Lemma 2.5.1, $\mathcal{G}$ is a valid (ε, δ)-DP privacy-loss accumulator. □

4

Concurrent Filters and Odometers for Rényi Divergence-Based DP Definitions

4.1 Rényi and Zero-Concentrated Differential Privacy

Rényi differential privacy, proposed first by Mironov,[31] is a relaxation of (ε, δ)-DP that better supports the composition of heterogeneous mechanisms. It is directly based off of Rényi divergence,

defined in Definition 1.1.9.

Definition 4.1.1 (Rényi-DP[31]). *An interactive mechanism $\mathcal{M}$ is* (α, ε)-Rényi differentially private *if for all every two neighboring datasets x and x' and every adversary $\mathcal{A}$,*

$$D_\alpha(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x)) || \texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x'))) \leq \varepsilon$$

where D_α is the Rényi divergence defined in Definition 1.1.9.

In our generalized DP framework in Definition 1.5.1, for Rényi-DP of order α, the partially-ordered set $(\mathcal{D}, \preceq)$ is $((\mathbb{R}^{\geq 0}) \cup \{\infty\}, \leq)$. The distance mapping $D(P, Q)$ for two probability distributions P and Q is α-Rényi divergence $D_\alpha(P, Q)$ for $\alpha \in (1, \infty)$. Rényi divergence is also closed under postprocessing due to the data-processing inequality, and it satisfies the joint convexity property.[38] We will notate RDP_α as the family of distance measures corresponding to Rényi-DP mechanisms of a fixed order α, i.e. any $(\alpha, \cdot)$-RDP mechanism would be RDP_α and vice versa.

Rényi-DP is a relaxation of (ε, δ)-DP that allows a more convenient way of tracking privacy guarantees of advanced composition theorems, which can become complicated with approximate-DP.[31] Rényi-DP is particularly useful for succinctly capturing privacy guarantees for the Gaussian mechanism and describing its behavior under composition. The Gaussian mechanism is convenient because the sum of two independent Gaussians is a Gaussian, so noise added for privacy may be the same as the error that was initially in the dataset.[15,31] The tails of the Gaussian mechanism also decay a lot faster than the tails of the Laplace mechanism, another noising mechanism.[31] However, the Gaussian mechanism cannot be captured by ε-DP, and satisfies a range of (ε, δ)-DP guarantees, making it difficult to represent concisely the privacy guarantee of the Gaussian mechanism using either of these definitions.[31] In comparison, Rényi-DP is much more convenient.

Example 4.1.2 (Gaussian mechanism is $\frac{\alpha}{2\sigma^2}$-RDP_α.[5,10]). *For any pair of neighboring datasets x, x'*

and a function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$ *with* ℓ_2 *sensitivity* 1*, the Gaussian mechanism satisfies* $\frac{\alpha}{2\sigma^2}$*-RDP*$_\alpha$ *for any* $\alpha > 1$*.*

Researchers have also shown that the same holds for discrete Gaussian with ℓ_2 sensitivity 1.[6]

Another definition of differential privacy derived from Rényi divergence is zero-concentrated differential privacy.

Definition 4.1.3 (Zero-concentrated differential privacy (zCDP)[5])**.** *An interactive mechanism* $\mathcal{M}$ *is* ρ-zero-concentrated differentially private (ρ-zCDP) *if for every two neighboring datasets* x *and* x'*, every adversary* $\mathcal{A}$*, and all orders* $\alpha \in (1, \infty)$*,*

$$D_\alpha(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x))||\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x'))) \leq \rho\alpha,$$

where D_α *is the Rényi divergence defined in Definition 1.1.9.*

In other words, $\mathcal{M}$ *is* ρ*-zCDP iff* $\mathcal{M}$ *is* $(\alpha, \rho\alpha)$*-RDP for every* $\alpha \in (1, \infty)$*.*

The privacy of the Gaussian mechanism follows directly from the definition of zCDP as a corollary.

Corollary 4.1.4 (Gaussian mechanism is zCDP[5])**.** *The Gaussian mechanism with parameter* σ^2*, on a function with sensitivity* 1*, is* $\frac{1}{2\sigma^2}$*-zCDP.*

Since zCDP bounds Rényi divergence for all orders α, it is stronger than Rényi-DP, which bounds the divergence for just one order.

In our generalized probability framework in Definition 1.5.1, the partially ordered set $\mathcal{D}$ for zCDP is $((\mathbb{R}^{\geq 0} \cup \{\infty\}, \leq)$. The distance mapping is the worst-case Rényi divergence normalized by α, over all $\alpha \in (1, \infty)$. In other words, $D(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x))||\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x'))) = \sup_{\alpha \in (1,\infty)} \left(\frac{D_\alpha(\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x))||\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{M}(x')))}{\alpha} \right)$. zCDP is closed under postprocessing[5,27] and satisfies the joint convexity property as well.[5]

Similar to Rényi-DP, the concurrent composition of zCDP mechanisms is additive.

Theorem 4.1.5 (Concurrent composition of zCDP mechanisms[27]). *For $k \in \mathbb{N}, \rho_1, \ldots, \rho_k > 0$, and all interactive mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_k$ such that $\mathcal{M}_i$ is ρ_i-zCDP, their concurrent composition $ConComp(\mathcal{M}_1, \ldots, \mathcal{M}_k)$ is $\sum_{i=1}^{k} \rho_i$-zCDP.*

Rényi-DP and zCDP both support a tighter analysis of the Gaussian mechanism than approximate-DP would. One way to think about these definitions of DP is in terms of a *privacy-loss random variable*,[15] which is a random variable that refers to the privacy loss at any point in time, as defined in Definition 1.1.1. The advantage of ε-DP is that its privacy parameters are additive, and while the relaxation of approximate-DP allows for tighter bounds for the composition of ε-DP mechanisms, iterating this process leads to combinatorially many cases due to the fact that there can be multiple (ε, δ) privacy guarantees for a given scenario.[31] In contrast, Rényi-DP and zCDP both bound the moment of this random variable. Rényi-DP bounds one particular moment, while zCDP bounds all positive moments. This allows for tackling the problem of composing mechanisms that satisfy several (ε, δ) guarantees simultaneously. Hence, both definitions capture more information more concisely under composition compared to approximate-DP.

In this chapter, we will present a Rényi-DP concurrent filter with a continuation rule that corresponds to the concurrent composition of Rényi-DP, where the privacy-loss parameters are specified upfront, proved independently by Lyu[27] and Vadhan and Zhang[38] and given in Theorem 4.1.6 below. Our proof for a concurrent Rényi-DP filter leverages the $K = 2$ case of this concurrent composition theorem.

Theorem 4.1.6 (Concurrent composition of Rényi-DP mechanisms[27,38]). *For all $\alpha > 1$, $k \in \mathbb{N}$, $\varepsilon_1, \ldots, \varepsilon_k > 0$, and all interactive mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_k$ such that $\mathcal{M}_i$ is ε_i-RDP$_\alpha$, the concurrent composition $ConComp(\mathcal{M}_1, \ldots, \mathcal{M}_k)$ is $\sum_{i=1}^{k} \varepsilon_i$-RDP$_\alpha$.*

We will see that once we have a concurrent Rényi-DP filter, the concurrent odometer case follows directly. Concurrent filters and odometers for zCDP will follow directly from the Rényi filter result as well.

4.2 Concurrent Filter for Rényi-DP

It turns out that even though privacy parameters are adaptively chosen in the filter setting, it is sufficient to use the summation of privacy parameters for the continuation rule of a concurrent filter.

Theorem 4.2.1 (Concurrent filter for Rényi-DP mechanisms)**.** *An RDP filter with the continuation rule* $\mathcal{F}(\varepsilon_1, \varepsilon_2, \ldots; \varepsilon) = \mathbb{I}(\sum_i \varepsilon_i \leq \varepsilon)$ *is a valid RDP IM-filter for every fixed order of* $\alpha > 1$.

Though generalized probability distance also covers RDP, the reduction in Theorem 3.1.4 holds only if the probability distance also satisfies properties of coupling and chain rule,[38] which RDP does not. Therefore, we use a different strategy to prove 4.2.1. We prove this theorem by inducting on a finite number of mechanisms being maintained by the $\mathcal{F}$-filtered concurrent composition of interactive mechanisms $\mathcal{F}$-*FiltCon*(IM), and then taking the limit as the number of mechanisms tends to ∞. To do so, we first consider a $\mathcal{F}_K$-*FiltCon*(IM) that maintains up to K mechanisms.

Lemma 4.2.2. *Let* $\alpha > 1$ *and define*

$$\mathcal{F}_K(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_k; \varepsilon) = \begin{cases} \mathbb{I}(\sum_{i=1}^{k} \varepsilon_i \leq \varepsilon) & \text{if } k \leq K \\ 0 & \text{otherwise} \end{cases}$$

Then $\mathcal{F}_K$ *is a valid* RDP_α *IM-filter.*

Proof. We prove this by inducting on the maximum number K of mechanisms that $\mathcal{F}_K$-*FiltCon*(IM) maintains. Fix a pair of datasets x, x'. By Lemma 1.5.5, it suffices to consider a deterministic adversary to prove this theorem.

Base case: $K = 2$. Intuitively, with two mechanisms, the budget partition is determined before any mechanisms are queried, so we can reduce the problem to concurrent RDP composition. Suppose the adversary starts a ε_1-RDP$_\alpha$ mechanism $\mathcal{M}_1$ through $\mathcal{F}_2$-*FiltCon*(IM). By the constraints of $\mathcal{F}_2$-*FiltCon*(IM), the second mechanism $\mathcal{M}_2$ it is able to start must have privacy parameter $\varepsilon_2 \leq \varepsilon - \varepsilon_1$. The adversary can adaptively interleave queries between $\mathcal{M}_1$ and $\mathcal{M}_2$.

We will first define the notion of a *universal* mechanism to help us with the reduction. Informally, a "universal" mechanism can be any mechanism with the privacy loss bounded by the preset privacy loss budget. The universal mechanism will first verify if an $\mathcal{M}$ attempted by an adversary to start satisfies the given privacy loss parameter guarantee as well as if the given privacy loss parameter does not exceed the preset privacy loss budget. Upon verifying this, it will interact with the adversary as $\mathcal{M}$. Given a preset privacy budget ε, an RDP$_\alpha$ universal mechanism $\mathcal{U}(s, m)$ would operate as follows:

1. If $m = \lambda$, initialize $s = (x, ())$ and return (s, λ)
2. If $m = (\mathcal{M}, \varepsilon_1)$, where $\mathcal{M}$ is an RDP$_\alpha$ mechanism:
    (a) If $s = (x, ())$ and $\varepsilon_1 \leq \varepsilon$, let $s' = (x, (\mathcal{M}, \varepsilon_1, s_\mathcal{M}))$ and $m' =$ `yes`
    (b) Else let $s' = s$ and $m' =$ `invalid query`
3. If $m = q$, where q is a query:
    (a) If $s = (x, ())$, let $s' = s$ and $m' =$ `no mechanism to query`
    (b) Else parse $s = (x, (\mathcal{M}, \varepsilon_1, s_\mathcal{M}))$:
        i. Let $(s'_\mathcal{M}, m') = \mathcal{M}(s_\mathcal{M}, q)$
        ii. Let $s' = (x, (\mathcal{M}, \varepsilon_1, s'_\mathcal{M}))$
4. Return (s', m')

Given an adversary $\mathcal{A}$ and an interaction between $\mathcal{A}$ and $\mathcal{F}_2$-*FiltCon*(IM), we will construct an adversary $\mathcal{A}'$ that interacts with $\text{ConComp}(\mathcal{M}_1, \mathcal{U}_2)$, where $\mathcal{U}_2$ is a universal $(\varepsilon - \varepsilon_1)$-RDP$_\alpha$ mechanism, and the adversary's first query to $\mathcal{U}_2$ is an ε_2-RDP$_\alpha$ mechanism $\mathcal{M}_2$. For each query $\mathcal{A}$ makes to $\mathcal{F}_2$-*FiltCon*(IM) prior to starting $\mathcal{M}_2$, $\mathcal{A}'$ will make a query to $\mathcal{M}_1$. When $\mathcal{A}$ starts $\mathcal{M}_2$, $\mathcal{A}'$ will make its first query to $\mathcal{U}_2$. Upon verifying $\varepsilon_2 \leq \varepsilon - \varepsilon_1$, $\mathcal{U}_2$ interacts with the adversary just as $\mathcal{M}_2$ does. Afterward, $\mathcal{A}'$ can interleave queries at will between $\mathcal{M}_1$ and $\mathcal{M}_2$, interacting with $\text{ConComp}(\mathcal{M}_1, \mathcal{U}_2)$ just as $\mathcal{A}$ does with $\mathcal{F}_2$-*FiltCon*(IM).

In this case, $\texttt{View}(\mathcal{A}' \leftrightarrow \text{ConComp}(\mathcal{M}_1, \mathcal{U}_2)(x)) \equiv \texttt{View}(\mathcal{A}' \leftrightarrow \text{ConComp}(\mathcal{M}_1, \mathcal{M}_2)(x))$, and the same holds for the computation on x'. By Theorem 4.1.6, $\text{ConComp}(\mathcal{M}_1, \mathcal{M}_2)$ is ε-RDP$_\alpha$, which means $\mathcal{F}_2$-*FiltCon*(IM) an ε-RDP$_\alpha$ I.M. By Definition 2.2.3, $\mathcal{F}_2(\cdot; \varepsilon)$ is a valid RDP$_\alpha$ IM-filter.

Induction on K. We assume that $\mathcal{F}_{K-1}$-*FiltCon*(IM) is differentially private. We define the post-processing $\mathcal{P}$ interacting with $\mathcal{F}_2$-*FiltCon*(IM) so that for any $K > 1$ and dataset x, $\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{P} \circ \mathcal{F}_2\text{-}\textit{FiltCon}(IM)(x)) \equiv \texttt{View}(\mathcal{A} \leftrightarrow \mathcal{F}_K\text{-}\textit{FiltCon}(IM)(x))$. The algorithmic pseudocode of $\mathcal{P}$ is in Algorithm 13, and the schema is depicted in Figure 4.1.

This algorithm assumes inductively that $\mathcal{F}_{K-1}$-*FiltCon*(IM) is a valid RDP$_\alpha$ filter. For any adversary, when receiving the adversary's request to start $\mathcal{M}_1$ and verifying the privacy loss is within budget, $\mathcal{P}$ will start both $\mathcal{M}_1$ and $\mathcal{F}_{K-1}$-*FiltCon*(IM) in $\mathcal{F}_2$-*FiltCon*(IM). $\mathcal{P}$ will directly pass on queries to $\mathcal{M}_1$. For queries q to $\mathcal{M}_j$ where $j > 1$ (in the form (j, q)), $\mathcal{P}$ will postprocess the message as $(2, (j-1), q)$, indicating that $\mathcal{P}$ is passing q on to the $(j-1)$st mechanism of $\mathcal{F}_{K-1}$-*FiltCon*(IM). (Concretely, suppose the adversary thinks it's interacting concurrently with $\mathcal{F}_4$-*FiltCon*(IM) maintaining $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4$. In this construction, the adversary will actually be interacting with $\mathcal{F}_2$-*FiltCon*(IM) maintaining $\mathcal{M}_1$ and $\mathcal{F}_3$-*FiltCon*(IM), the latter of which is comprised of $\mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4$, which are the first, second, and third mechanisms, respectively, of $\mathcal{F}_3$-*FiltCon*(IM).)

Algorithm 13 Interactive postprocessing $\mathcal{P}$ that transforms $\mathcal{F}_2$-*FiltCon*(IM) into $\mathcal{F}_K$-*FiltCon*(IM).

procedure $\mathcal{P}(s, v, m; r)$:
  $k \leftarrow$ number of mechanisms (started or attempting to start)
  $K \leftarrow$ maximum number of mechanisms
  if $m = \lambda$ **then** ▷ initialize $\mathcal{P}$
    $s' \leftarrow \lambda, v \leftarrow A$
    return (s', v, λ)
  end if
  if $v = Q$ **then** ▷ $\mathcal{P}$ accepting queries
    if $m = (\mathcal{M}', (\alpha, \varepsilon'))$ **then** ▷ $\mathcal{A}$ starts a new mechanism
      if $k = 0$ **then** ▷ currently maintaining no mechanisms
        $\mathcal{M}_1 \leftarrow \mathcal{M}', \varepsilon_1 \leftarrow \varepsilon'$
        $b_1 \leftarrow$ no ▷ b_1 is the boolean keeping track of whether $\mathcal{M}_1$ has successfully started
        $s^{\mathcal{P}'} \leftarrow ((\mathcal{M}_1, (\alpha, \varepsilon_1), b_1))$
      else
        $s^{\mathcal{P}'} \leftarrow s^{\mathcal{P}}$
      end if
      $m' \leftarrow m$ ▷ pass on the query; the filter will handle any rejections
    else if $m = (j, q)$ where $j = 1, \ldots, K$ and q is a query to $\mathcal{M}_j$ **then** ▷ $\mathcal{A}$ queries an old mechanism
      if $j = 1$ **then** ▷ querying $\mathcal{M}_1$
        $s^{\mathcal{P}'} \leftarrow s^{\mathcal{P}}, m' \leftarrow m$
      else ▷ querying anything other than $\mathcal{M}_1$
        $m' \leftarrow (2, (j-1, q))$
        $s^{\mathcal{P}'} \leftarrow s^{\mathcal{P}}, m' \leftarrow m$
      end if
    end if
    $v \leftarrow A$ ▷ we will be talking to the mechanism next
    return $(s^{\mathcal{P}'}, Q, m')$
  else if $v = A$ **then** ▷ $\mathcal{P}$ accepting answers
    if $m =$ yes and $k = 1$ **then** ▷ successfully started $\mathcal{M}_1$
      $b_1 \leftarrow$ yes
      $m' \leftarrow (\mathcal{F}_{K-1}\text{-}FiltCon(IM), (\alpha, \varepsilon - \varepsilon_1))$
      $s^{\mathcal{P}'} \leftarrow ((\mathcal{M}_1, (\alpha, \varepsilon_1), b_1), (\mathcal{F}_{K-1}\text{-}FiltCon(IM), (\alpha, \varepsilon - \varepsilon_1), \text{no}))$ ▷ start $\mathcal{F}_{K-1}$-*FiltCon*(IM) immediately
      return $(s^{\mathcal{P}'}, Q, m')$ ▷ pass on the query
    else
      $s^{\mathcal{P}'} \leftarrow s^{\mathcal{P}}, v \leftarrow Q, m' \leftarrow m$
      return $(s^{\mathcal{P}'}, A, m')$
    end if
  end if
end procedure

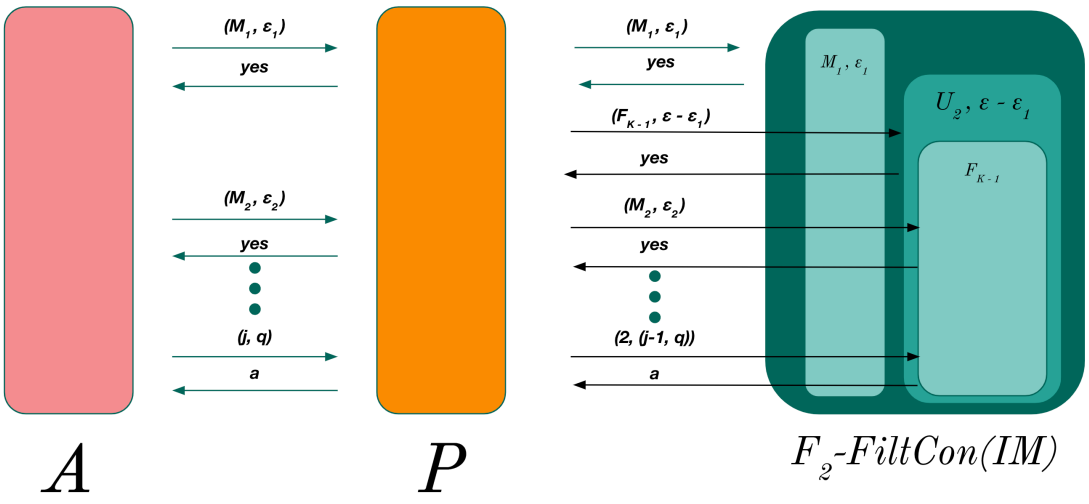


Figure 4.1: Schematic of IM-to-IM postprocessing $\mathcal{P}$ for Theorem 4.2.1. $\mathcal{P}$ starts $\mathcal{M}_1$ and $\mathcal{F}_{K-1}$ together following the adversary's request to start $\mathcal{M}_1$, and passes queries to mechanisms that are not to $\mathcal{M}_1$ to their corresponding mechanisms in $\mathcal{F}_{K-1}$.

Given a privacy-loss budget ε, suppose the adversary starts a ε_1-RDP$_\alpha$ mechanism $\mathcal{M}_1$ with $\varepsilon_1 \leq \varepsilon$. Inductively assume $\mathcal{F}_{K-1}$-*FiltCon*$(IM)(x, \cdot)$ is an $(\varepsilon - \varepsilon_1)$-RDP$_\alpha$ I.M. As soon as $\mathcal{M}_1$ gets created, $\mathcal{P}$ will feed in $\mathcal{M}_1$ to $\mathcal{F}_2$-*FiltCon*(IM). Upon receiving confirmation that $\mathcal{M}_1$ has properly started, $\mathcal{P}$ will then feed in $\mathcal{F}_{k-1}$ to $\mathcal{F}_2$-*FiltCon*(IM) with an $\varepsilon - \varepsilon_1$ budget. We don't need to handle the parsing in the $v = Q$ section because $\mathcal{P}$ will always start two mechanisms at once. We can view $\mathcal{F}_{K-1}$-*FiltCon*(IM) as the first query to $\mathcal{U}_2$.

By construction, for any deterministic adversary $\mathcal{A}$ and dataset x, $\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{P} \circ \mathcal{F}_2\text{-}FiltCon(IM)(x))$ is identically distributed as $\texttt{View}(\mathcal{A} \leftrightarrow \mathcal{F}_K\text{-}FiltCon(IM)(x))$. By Theorem 2.4.3, $\mathcal{P} \circ \mathcal{F}_2$-*FiltCon*$(IM)$ is an ε-RDP$_\alpha$ I.M., which means $\mathcal{F}_K$-*FiltCon*(IM) is ε-RDP$_\alpha$ I.M. By Definition 2.2.3, $\mathcal{F}_K$ is a valid RDP$_\alpha$ IM-filter. □

To show that the filter works with an unbounded number of mechanisms, we examine the behavior as $K \to \infty$. We define a sequence of measurable spaces that are well-behaved in the sense that we can safely marginalize out distributions even when the sequence is not finite. Suppose $(\Omega^\infty, \Gamma^\infty)$ is

the direct product of an infinite sequence of measurable spaces $(\Omega_1, \Gamma_1), (\Omega_2, \Gamma_2), \ldots$, i.e. $\Omega^\infty = \Omega_1 \times \Omega_2 \times \ldots$ and Γ^∞ is the smallest σ-algebra containing all cylinder sets $S_n(A) = \{x^\infty \in \Omega^\infty \mid x_1, x_2, \ldots, x_n \in A\}, A \in \Gamma^n$, for $n = 1, 2, \ldots$ and $\Gamma^n = \Gamma_1 \otimes \Gamma_2 \otimes \cdots \otimes \Gamma_n$. We call a sequence of probability distributions $P^1, P^2, \ldots$ *consistent* if P^n is a distribution on $\Omega^n = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_n$ and $P^{i+1}(A \times \Omega_{i+1}) = P^i(A)$ for $A \in \Gamma^i$, $i = 1, \ldots, n$.

For any such sequence there exists a distribution P^∞ on $(\Omega^\infty, \Gamma^\infty)$ such that its marginal distribution on Ω^n is P^n, in the sense that $P^\infty(S_n(A)) = P^n(A), A \in \Gamma^n$.

Lemma 4.2.3 (Rényi divergence of the limit of a sequence of probability distributions[39])**.** *Let $P^1, P^2, \ldots$ and $Q^1, Q^2, \ldots$ be consistent sequences of probability distributions on $(\Omega^1, \Gamma^1), (\Omega^2, \Gamma^2), \ldots$, where, for $n = 1, 2, \ldots, \infty$, (Ω^n, Γ^n) is the direct product of the first n measurable spaces in the infinite sequence $(\Omega_1, \Gamma_1), (\Omega_2, \Gamma_2), \ldots$. Then for any $a \in (0, \infty]$*

$$\lim_{n\to\infty} D_a(P^n||Q^n) = D_a(P^\infty||Q^\infty).$$

Proof of Theorem 4.2.1. Fix a pair of datasets x, x' and a divergence parameter $\alpha > 1$. Consider a $\mathcal{F}_k(\cdot;\varepsilon)$-*FiltCon*$(IM)$ with exactly k queries in its `View`. (Note that this implies the adversary interacts with k queries at maximum, keeping within the constraints of the concurrent filter.) Let q_i denote the adversary's ith query and a_i denote the answer to the adversary's ith query.

Define sequences of random variables $X_1, \ldots, X_k$ and $Y_1, \ldots, Y_k$ where $X_i := (q_1, a_1, \ldots, q_i, a_i)$ on x and $Y_i := (q_1, a_1, \ldots, q_i, a_i)$ on x'. The corresponding distributions for X_i, Y_i are X^i, Y^i respectively. By construction, both the sequences of X_i and Y_i are consistent.

By Theorem 4.1.6, for every $k = 1, 2, \ldots$ and $\alpha > 1$, we have $D_\alpha(X_k||Y_k) \leq \sum_{i=1}^k \varepsilon_i$, where equality holds if each query starts a new mechanism. By construction, $\sum_{i=1}^k \varepsilon_i \leq \varepsilon$. Then, invoking Lemma 4.2.3,

$$D_\alpha(X_\infty||Y_\infty) = \lim_{k\to\infty} D_\alpha(X_k||Y_k) \leq \varepsilon,$$

where the last inequality is because the sequence of $D_\alpha(X_k||Y_k)$ monotonically increases as $k \to \infty$, and is bounded above by ε. □

Our approach can be used to simplify Feldman and Zrnic's proof[18] for the bounds on privacy filters for sequentially-composed noninteractive DP mechanisms. They identify the sequence of losses incurred for each query – in this case the invocation of a noninteractive mechanism and the reception of its answers – as a supermartingale. They then apply the optional stopping theorem for martingales to bound the divergence between adjacent datasets and verify the validity of the filter.

The basis for our proof is that concurrent composition of two interactive RDP mechanisms implies a fully adaptive $\mathcal{F}_2$-*FiltCon*(IM). Having interactivity allows us to proceed by induction in the proof of Lemma 4.2.2, describing $\mathcal{F}_K$ in terms of $\mathcal{F}_{K-1}$ for some arbitrary K. We could do the same for the sequential composition since induction preserves sequentiality. Then by Lemma 4.2.2, $\mathcal{F}_K$-*FiltSeq*(IM) is an ε-RDP$_\alpha$ I.M. Thus, the sequential composition of two interactive Rényi-DP mechanisms of order α implies a fully adaptive filter for sequential composition of RDP mechanisms of order α.

4.3 Concurrent Odometer for Rényi-DP

Because of the bijective relationship between valid continuation rules and privacy-loss accumulators, we are able to define an additive relationship for concurrent odometers of Rényi mechanisms as well.

Theorem 4.3.1 (Valid ε-RDP$_\alpha$ odometer for concurrently-composed interactive mechanisms)**.** *Let $\alpha > 1$. Let $\mathcal{G} : \mathcal{D}^* \to \mathcal{D}$ be a relation. If $\mathcal{G}(\varepsilon_1, \varepsilon_2, \dots) = \sum_{i=1} \varepsilon_i$, then $\mathcal{G}$ is a valid privacy-loss accumulator for the concurrent composition of RDP$_\alpha$ I.M.s.*

Proof. Define a relation $\mathcal{F}(\varepsilon_1, \varepsilon_2, \dots; \varepsilon) : \mathcal{D}^* \times \mathcal{D} \to \mathcal{D}$ such that $\mathcal{F}(\varepsilon_1, \varepsilon_2, \dots; \varepsilon) = \mathbb{I}(\mathcal{G}(\varepsilon_1, \varepsilon_2, \dots) \leq$

ε). By Theorem 4.2.1, $\mathcal{F}$ is a valid RDP$_\alpha$ filter. By Lemma 2.5.1, $\mathcal{G}$ is a valid privacy-loss accumulator for Rényi mechanisms of order α. □

4.4 Concurrent Filter & Odometer for zCDP

Theorem 4.4.1 (Valid zCDP filter for concurrently-composed interactive mechanisms)**.** *Let $\mathcal{F} : \mathcal{D}^* \times \mathcal{D} \to \mathcal{D}$ be a relation. If $\mathcal{F}(\rho_1, \rho_2, \ldots; \rho) = \mathbb{I}(\sum_i \rho_i \leq \rho)$, then $\mathcal{F}$ is a valid ρ-zCDP filter for the concurrent composition of zCDP interactive mechanisms.*

Proof. By Definition 4.1.3, a mechanism $\mathcal{M}_i$ that is ρ_i-zCDP is also $(\alpha, \rho_i\alpha)$-RDP for all $\alpha > 1$. For a particular α, by Theorem 4.2.1, $\mathbb{I}(\sum_i \rho_i\alpha \leq \rho\alpha) = \mathbb{I}(\alpha \sum_i \rho_i \leq \rho\alpha) = \mathbb{I}(\sum_i \rho_i \leq \rho)$ is a valid RDP IM-filter, meaning that a filter $\mathcal{F}^{(\alpha)}$-*FiltCon*(IM) constructed from $\mathcal{F}^{(\alpha)}(\rho_1\alpha, \rho_2\alpha, \ldots; \rho\alpha) := \mathbb{I}(\sum_i \rho_i\alpha \leq \rho\alpha) = \mathcal{F}(\rho_1, \rho_2, \ldots; \rho)$ is an $(\alpha, \rho\alpha)$-RDP I.M. Since Theorem 4.2.1 holds for all α, $\mathcal{F}^{(\alpha)}$-*FiltCon*(IM) is an $(\alpha, \rho\alpha)$-RDP I.M. for all $\alpha > 1$, meaning that $\mathcal{F}(\rho_1, \rho_2, \ldots; \rho) = \mathbb{I}(\sum_i \rho_i \leq \rho)$ is a valid ρ-zCDP IM-filter. □

Theorem 4.4.2 (Valid ρ-zCDP odometer for concurrently-composed interactive mechanisms)**.** *Let $\mathcal{G} : \mathcal{D}^* \to \mathcal{D}$ be a relation. If $\mathcal{G}(\rho_1, \rho_2, \ldots) = \sum_{i=1} \rho_i$, then $\mathcal{G}$ is a valid privacy-loss accumulator for the concurrent composition of a sequence of mechanisms $\mathcal{M}_1, \mathcal{M}_2, \ldots$, where $\mathcal{M}_i$ is an ρ_i-zCDP interactive mechanism.*

Proof. Define a relation $\mathcal{F}(\rho_1, \rho_2, \ldots; \rho) : \mathcal{D}^* \times \mathcal{D} \to \mathcal{D}$ such that $\mathcal{F}(\rho_1, \rho_2, \ldots; \rho) = \mathbb{I}(\mathcal{G}(\rho_1, \rho_2, \cdots \leq \rho)$. By Theorem 4.4.1, $\mathcal{F}$ is a valid ρ-zCDP filter. By Lemma 2.5.1, $\mathcal{G}$ is a valid zCDP privacy-loss accumulator. □

Uncertainty is not chaos but rather the necessary habitat

of the present tense.

Shoshanna Zuboff, *The Age of Surveillance Capitalism*

5

Conclusion

In this thesis, we examined the adaptive concurrent composition of interactive mechanisms for approximate-DP, *f*-DP, Rényi-DP, and zCDP. In Chapter 2, we devised primitives for concurrent filters and odometers and proved that our formulation allows an analyst to convert between the two. In Chapter 3, we extended these primitives to the approximate-DP and *f*-DP settings, and in Chapter 4, we extended these primitives to the Rényi-DP and zCDP settings. The technical contribution of this thesis lies in building intuitive privacy-loss tracking primitives that can be adapted for future

work and are easily translatable into code. We also show that current theorems for fully-adaptive composition of non-interactive mechanisms still apply to the fully-adaptive, concurrent composition of interactive mechanisms, which is convenient for practical libraries.

5.1 Practical Implications

The results of this thesis have a number of practical implications. The analyst can now engage with datasets and mechanisms in a more exploratory way, by partitioning its privacy budget freely across mechanisms, instead of having to decide on a budget partition ahead of time. This way, if while interacting with mechanisms the analyst changes their mind about how much privacy budget they want to allocate to a subsequent mechanism, they can do so with impunity.

The theorems proved in this thesis will also streamline implementations and reduce code complexity. Many DP repositories, such as Tumult Labs[25] or the open-source OpenDP,[19] currently require complicated sequentiality constraints, to prevent the analyst from interleaving queries between mechanisms and potentially causing a privacy violation. In particular, implementations previously had to enforce constraints against concurrent composition in scenarios in which it was not allowed. As this thesis confirms the privacy of concurrent composition with adaptive parameters, the repositories can now do away with such sequentiality constraints.

5.2 Future Directions

Future theoretical directions concern more abstract odometers. Not all odometers need to be equipped with a privacy-loss accumulator, and not all odometers need to be built from the adaptive concurrent composition of I.M.s. Some odometers may simply answer a `privacy_loss` query from its current state. Generalizing our theorems to these settings could pave the way for theorems of a broader scope.

Odometers of odometers are particularly useful in the case that an analyst would like to compose multiple odometers concurrently, hence allowing the analyst to not only track the privacy loss in the computation in total but also the privacy loss in each individual computation as composed. Our current theorems support only odometers that are built from privacy-loss accumulators, made from concurrently-composed I.M.s; because an odometer of odometers is a more general setting, handling its properties is a bit more nuanced of a task. The adversary's objective is now no longer just to maximize divergence; the adversary must also strategize to get the divergence to exceed the response of a `privacy_loss` query.

Another further direction is extending our concurrent odometers to other forms of DP, such as approximate-zCDP,[5] which currently has no known concurrent composition theorems with pre-fixed privacy parameters. Having provable privacy guarantees for even more variants of DP is always desirable for utility.

Ultimately, by expanding the set of real-world settings to which DP primitives are extensible, we hope to reduce one barrier to the widespread adoption of DP frameworks.

References

[1] Anthony, A. (2023). Cyber resilience must focus on marginalized individuals, not just institutions.

[2] Balcer, V. and Vadhan, S. (2019). Differential privacy on finite computers.

[3] Beaudry, N. J. and Renner, R. (2012). An intuitive proof of the data processing inequality.

[4] Berghel, S., Bohannon, P., Desfontaines, D., Estes, C., Haney, S., Hartman, L., Hay, M., Machanavajjhala, A., Magerlein, T., Miklau, G., Pai, A., Sexton, W., and Shrestha, R. (2022). Tumult Analytics: a robust, easy-to-use, scalable, and expressive framework for differential privacy. *arXiv preprint arXiv:2212.04133*.

[5] Bun, M. and Steinke, T. (2016). Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pages 635–658. Springer.

[6] Canonne, C. L., Kamath, G., and Steinke, T. (2021). The discrete gaussian for differential privacy.

[7] Casacuberta, S., Shoemate, M., Vadhan, S., and Wagaman, C. (2022). Widespread underestimation of sensitivity in differentially private libraries and how to fix it. CCS '22, page 471–484, New York, NY, USA. Association for Computing Machinery.

[8] Chao, B., Null, E., Collins-Dexter, B., and Park, C. (2019). https://www.newamerica.org/oti/reports/centering-civil-rights-privacy-debate/.

[9] Dinur, I. and Nissim, K. (2003). Revealing information while preserving privacy. In *Proceedings of the Twenty-Second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '03, page 202–210, New York, NY, USA. Association for Computing Machinery.

[10] Dong, J., Roth, A., and Su, W. J. (2019). Gaussian differential privacy. *arXiv preprint arXiv:1905.02383*.

[11] Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. (2006a). Our data, ourselves: Privacy via distributed noise generation. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 486–503. Springer.

[12] Dwork, C. and Lei, J. (2009). Differential privacy and robust statistics. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, page 371–380, New York, NY, USA. Association for Computing Machinery.

[13] Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006b). Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, TCC '06, pages 265–284.

[14] Dwork, C., Naor, M., Reingold, O., Rothblum, G. N., and Vadhan, S. P. (2009). On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proceedings of the 41st ACM Symposium on Theory of Computing*, STOC '09, pages 381–390.

[15] Dwork, C. and Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407.

[16] Dwork, C. and Rothblum, G. N. (2016). Concentrated differential privacy. *arXiv preprint arXiv:1603.01887*.

[17] Dwork, C., Rothblum, G. N., and Vadhan, S. (2010). Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60.

[18] Feldman, V. and Zrnic, T. (2022). Individual privacy accounting via a renyi filter. *arXiv preprint arXiv:2008.11193*.

[19] Gaboardi, M., Hay, M., and Vadhan, S. (2020). A programming framework for opendp. *Manuscript*.

[20] Ghosh, A., Roughgarden, T., and Sundararajan, M. (2009). Universally utility-maximizing privacy mechanisms.

[21] Hardt, M. and Rothblum, G. N. (2010). A multiplicative weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st annual symposium on foundations of computer science*, pages 61–70. IEEE.

[22] Kairouz, P., Oh, S., and Viswanath, P. (2015). The composition theorem for differential privacy. In *International conference on machine learning*, pages 1376–1385. PMLR.

[23] Kifer, D. and Machanavajjhala, A. (2011). No free lunch in data privacy. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, SIGMOD '11, page 193–204, New York, NY, USA. Association for Computing Machinery.

[24] Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

[25] Labs, T. (2022). Tumult Analytics.

[26] Lyu, M., Su, D., and Li, N. (2016). Understanding the sparse vector technique for differential privacy.

[27] Lyu, X. (2022). Composition theorems for interactive differential privacy. In *Thirty-sixth Conference on Neural Information Processing Systems*.

[28] Lécuyer, M. (2021). Practical privacy filters and odometers with rényi differential privacy and applications to differentially private deep learning.

[29] McSherry, F. and Talwar, K. (2007). Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103.

[30] Mironov, I. (2012). On significance of the least significant bits for differential privacy. CCS '12, page 650–661, New York, NY, USA. Association for Computing Machinery.

[31] Mironov, I. (2017). Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275. IEEE.

[32] Narayanan, A. and Shmatikov, V. (2008). Robust de-anonymization of large sparse datasets. In *Proceedings - 2008 IEEE Symposium on Security and Privacy, SP*, Proceedings - IEEE Symposium on Security and Privacy, pages 111–125. 2008 IEEE Symposium on Security and Privacy, SP ; Conference date: 18-05-2008 Through 21-05-2008.

[33] Rényi, A. (1961). On measures of entropy and information. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, volume 1. Berkeley, California, USA.

[34] Rogers, R., Roth, A., Ullman, J., and Vadhan, S. (2021). Privacy odometers and filters: Pay-as-you-go composition.

[35] Sunitha, B. and Kiran, B. K. (2023). A systematic review of recommendation systems: Applications and challenges. *International Journal of Intelligent Systems and Applications in Engineering*, 11(2):826–846.

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

[37] Vadhan, S. and Wang, T. (2021). Concurrent composition of differential privacy. In *Theory of Cryptography Conference*, pages 582–604. Springer.

[38] Vadhan, S. and Zhang, W. (2023). Concurrent composition theorems for differential privacy. In *55th Annual ACM Symposium on Theory of Computing*.

[39] Van Erven, T. and Harremos, P. (2014). Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820.

[40] Wasserman, L. and Zhou, S. (2010). A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389.

[41] Whitehouse, J., Ramdas, A., Rogers, R., and Wu, Z. S. (2023). Fully adaptive composition in differential privacy.

[42] Zhu, K., Fioretto, F., and Hentenryck, P. V. (2022). Post-processing of differentially private data: A fairness perspective.