Novel Approaches to Preserving Utility in Privacy Enhancing Technologies

Meisam Mohammady

A Thesis
in
The Concordia Institute
for
Information Systems Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy (Information and Systems Engineering) at
Concordia University
Montréal, Québec, Canada

September 2020

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: **Meisam Mohammady**

Entitled: **Novel Approaches to Preserving Utility in Privacy Enhancing Technologies**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Information and Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. Catherine Mulligan

_____ External Examiner
Dr. Bo Luo

_____ External to Program
Dr. Emad Shihab

_____ Internal Examiner
Dr. Jun Yan

_____ Internal Examiner
Dr. Jeremy Clark

_____ Thesis Supervisor
Dr. Lingyu Wang & Dr. Yuan Hong

Approved by _____
Dr. Mohammad Mannan, Graduate Program Director

September 30th, 2020 _____
Dr. Mourad Debbabi, Dean
Gina Cody School of Engineering and Computer Science

Abstract

Novel Approaches to Preserving Utility in Privacy Enhancing Technologies

Meisam Mohammady, Ph.D.

Concordia University, 2020

Significant amount of individual information are being collected and analyzed today through a wide variety of applications across different industries. While pursuing better utility by discovering knowledge from the data, an individual's privacy may be compromised during an analysis: corporate networks monitor their online behavior, advertising companies collect and share their private information, and cybercriminals cause financial damages through security breaches. To this end, the data typically goes under certain anonymization techniques, e.g., CryptoPAn [Computer Networks'04], which replaces real IP addresses with prefix-preserving pseudonyms, or *Differentially Private (DP)* [ICALP'06] techniques which modify the answer to a query by adding a zero-mean noise distributed according to, e.g., a Laplace distribution. Unfortunately, most such techniques either are vulnerable to adversaries with prior knowledge, e.g., some network flows in the data, or require heavy data sanitization or perturbation, both of which may result in a significant loss of data utility. Therefore, the fundamental trade-off between privacy and utility (i.e., analysis accuracy) has attracted significant attention in various settings [ICALP'06, ACM CCS'14]. In line with this track of research, in this dissertation we aim to build utility-maximized and privacy-preserving tools for Internet communications. Such tools can be employed not only by dissidents and whistleblowers, but also by ordinary Internet users on a daily basis. To this end, we combine the development of practical systems with rigorous theoretical analysis, and incorporate techniques from various disciplines such as computer networking, cryptography, and statistical analysis. During the research, we proposed three different frameworks in some well-known settings outlined in the following.

First, we propose *The Multi-view Approach* to preserve both privacy and utility in network trace anonymization, Second, *The R^2DP Approach* which is a novel technique on differentially private mechanism design with maximized utility, and Third, *The DPOD Approach* that is a novel framework on privacy preserving Anomaly detection in the outsourcing setting.

Acknowledgments

I would first like to thank my supervisors, Professor Lingyu Wang and Professor Yuan Hong, whose expertise were invaluable in formulating the research questions and methodology. Your insightful feedback pushed me to sharpen my thinking and brought my work to a higher level. I would like to acknowledge my colleagues Shangyu Xie, Han Wang, Prof. Suryadipta Majumdar, Dr. Mengyuan Zhang, Momen Oqaily and Dr. Yosr Jarrya for their wonderful collaboration. I would particularly like to single out my supervisor at Ericsson Inc., Professor Makan Pourzandi, I want to thank you for your patient support and for all of the opportunities I was given to further my research. In addition, I would like to express my gratitude towards the thesis committee members for examining my thesis. Finally, I could not have completed this dissertation without the support of my mother Nosrat, my sisters Shahnaz, Maryam and Delnia, and our sweet nephew Noyan, for their wise counsel and sympathetic ear. You are always there for me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

1.2 Motivation

In the last few years, various notion of privacy has emerged to provide formal privacy guarantees against adversaries with a variety of side information. For instance, Differential Privacy aims at limiting the risk enhancement to one's privacy when she/he contributes her/his data to a statistical database. This model ensures that adding or removing a single record does not significantly affect the outcome of the sanitized algorithm. However, such sanitized data often suffers from lack of sufficient utility when being used in different application. In this dissertation, we focus on such trade-off between privacy and utility in sanitizing an applications. These applications could vary from network traces, intelligent transportation systems, smart grids and smart buildings. In line with this track of research, in this dissertation we aim to build utility-maximized and privacy-preserving tools for Internet communications. Such tools can be deployed not only by dissidents and whistleblowers, but also by ordinary Internet users on a daily basis.

1.3 Objectives and Contributions

During my research, we have combined the development of practical systems with rigorous theoretical analysis, and incorporated techniques from various disciplines such as computer networking, cryptography, and statistical analysis. Specifically, we proposed three different frameworks in some well-known settings outlined in the following. The specific problems we explore in this dissertation include privacy preserving tool in both *local setting* (optimized application-aware mechanism design), and *outsourced setting* (network trace analysis and intrusion detection system).

The Multi-view Approach. In our first work, we proposed a novel technique called *Multi-view* to preserve both privacy and utility in network trace anonymization. Specifically, releasing the network data flows is very important to perform network research activities, and study the behavior of the network which is widely used in mitigating zero-day network attacks. However, organizations are usually reluctant to share their network traces due to privacy concerns over sensitive information, e.g., network and system configuration, which may potentially be exploited for attacks. In cases where data owners are convinced to share their network traces, the data are typically subjected to certain anonymization techniques, e.g., CryptoPAn, which replaces real IP addresses with prefix-preserving pseudonyms. However, most such techniques either are vulnerable to adversaries with prior knowledge about some network flows in the traces, or require heavy data sanitization or perturbation, both of which may result in a significant loss of data utility. In this work, we aim to preserve both privacy and utility through shifting the trade-off from between privacy and utility to between privacy and computational cost. The key idea is for the analysts to generate and analyze multiple anonymized views of the original network traces; those views are designed to be sufficiently indistinguishable even to adversaries armed with prior knowledge, which preserves the privacy, whereas one of the views will yield true analysis results privately retrieved by the data owner, which preserves the utility. We formally analyzed the privacy of our solution and experimentally evaluated it using real network traces provided by a major ISP. The results showed that our approach can significantly reduce the level of information leakage (e.g., less than 1% of the

information leaked by the state-of-the-art) with comparable utility.

The R^{2}DP Approach. In our second work, we proposed a novel technique on differentially private mechanism design with maximized utility. Specifically, since the meaning of data utility in different applications may vastly differ, a key challenge is to find the optimal randomization mechanism, i.e., the distribution and its parameters, for a given utility metric. Existing works have identified the optimal distributions in some special cases, while leaving all other utility metrics (e.g., machine learning and social networking utility metrics) as open problems. Since existing works mostly rely on manual analysis to examine the search space of all distributions, it would be an expensive process to repeat such efforts for each utility metric. To address such deficiency, this work proposes a novel approach that can automatically optimize different utility metrics found in diverse applications under a common framework. Our key idea comes from the known fact in probability theory that, by regarding the variance of the injected noise itself as a random variable, a two-fold distribution may approximately cover the search space of all distributions. Therefore, we can automatically find distributions in this search space to optimize different utility metrics in a similar manner, simply by optimizing the parameters of the two-fold distribution. Specifically, we define a universal framework, namely, *R*andomizing the *R*andomization mechanism of *D*ifferential *P*rivacy (R^2DP), and we formally analyze its privacy and utility. Our experiments showed that R^{2}DP can provide better results than the baseline distribution (Laplace) for several utility metrics with no known optimal distributions, whereas our results asymptotically approach to the optimality for utility metrics having known optimal distributions.

The DPOD Approach. In the final contribution, we propose a novel framework on privacy preserving anomaly detection which has numerous applications in a very wide variety of domains such as data cleaning, fraud detection, financial markets, intrusion detection, and law enforcement. As identifying anomalous activities grows more sophisticated in different applications, e.g., network security monitoring, IoT and online banking, there is an increasing need for outsourcing such tasks to third-party Managed Security Service Providers (MSSP) to benefit from a more effective solution (compared to in-house solution). While pursuing better utility by outsourcing such

tasks to MSSPs, individual's privacy may be compromised during an analysis. While differential privacy has emerged as a new paradigm to provide rigorous privacy protection by obscuring the presence or absence of individual records in a dataset, unfortunately, in some applications, such indistinguishability property of differential privacy is in direct contradiction with many applications, e.g., in anomaly detection which requires differentiating between anomalous and benign records. Specifically, popular approaches to differential privacy, such as the Laplace and exponential mechanisms, calibrate randomised smoothing through global sensitivity of the target non-private function. Bounding such sensitivity is often a prohibitively complex analytic calculation especially in identifying anomalous records, with relatively larger values for some features. As an alternative, we propose a sampler for estimating sensitivity of non-private mechanisms through a *no privacy for anomalous records* policy to significantly reduce the required distortion in providing a strong level of protection to records that are with high probability benign (results in weaker protection to those that are most likely malicious). Since this solution requires accessing to a reliable estimate of the distribution of the dataset to sample the sensitivity from (a naïve solution is to consider a uniform distribution as proposed by Rubinstein et al [152]), a key challenge is to iteratively and efficiently interact with the MSSP to construct the PDF of the sampler. Our solution for a data owner starts with naive solution (uniform distribution; or privacy for all records), and for the MSSP to agnostically learn/update a histogram distribution over the noisy data to enable probabilistically excluding the set of malicious records from the sensitivity sampler algorithm. We design and implement the *Differentially Private Outsourcing of Anomaly Detection (DPOD)* framework, and demonstrate on example learners how the DPOD approach adopts a naturally-relaxed privacy guarantee, while achieving significantly more accurate releases to enable identifying anomalous activities.

1.4 Backgrounds

We review some background on differential privacy for the theoretical foundations of the DPOD framework.

1.4.1 Differential Privacy

We follow the standard definitions of ϵ-differential privacy [57, 139]. Let D be a dataset of interest and d, d' be two adjacent subsets of D meaning that we can obtain d' from d simply by adding or subtracting the data of one individual. A randomization mechanism $\mathcal{M} : \mathsf{D} \times \Omega \rightarrow \mathsf{R}$ which is ϵ-differentially private, necessarily randomizes its output in such a way that for all $S \subset \mathsf{R}$,

$$\mathbb{P}(\mathcal{M}(d) \in S) \leq e^{\epsilon}\mathbb{P}(\mathcal{M}(d') \in S) \tag{1}$$

If the inequality fails, then at least a ϵ breach takes place, which means the difference between the prior distribution and posterior one is tangible. We recall below a basic mechanism that can be used to answer queries in an ϵ-differentially private way. We will only be concerned with queries that return numerical answers, i.e., a query is a mapping $q : \mathsf{D} \rightarrow \mathbb{R}$, where $\mathbb{R}$ is a set of real numbers. The following sensitivity concept plays an important role in the design of differentially private mechanisms [55].

Definition 1.4.1. *The sensitivity of a query $q : \mathsf{D} \rightarrow \mathsf{R}$ is defined as $\Delta q = \max_{d,d':Adj(d,d')} |q(d) - q(d')|$ [57, 139].*

While strong ϵ-DP is ideal, utility may demand compromise. In particular, (ϵ, γ)-Random Differential Privacy (RDP) offers an alternative relaxation, where the strong ϵ-DP holds on all but a small γ-proportion of unlikely database pairs.

Definition 1.4.2. *Randomized mechanism $\mathcal{M}_q : \mathsf{D} \times \Omega \rightarrow \mathbb{R}$ preserves (ϵ, γ)-random differential privacy, at privacy level $\epsilon > 0$ and confidence $\gamma \in (0, 1)$, if $\mathbb{P}(\forall S \subset \mathsf{R}, \mathbb{P}(\mathcal{M}(d) \in S) \leq e^{\epsilon} \cdot \mathbb{P}(\mathcal{M}(d) \in S)) \geq 1 - \gamma$, with the inner probabilities over the mechanism's randomization, and the outer probability over neighbouring $d, d' \in \mathsf{D}$ drawn from some P^{n+1}.*

1.4.2 Laplace Mechanism

The Laplace mechanism [55] modifies a numerical query result by adding zero-mean noise (denoted as $Lap(b)$) distributed according to a Laplace distribution with mean zero and scale parameter b. It has density $p(x; b) = \frac{1}{2b} exp(-\frac{|x|}{b})$ and variance $2b^2$.

Theorem 1.4.1. *Let $q : \mathsf{D} \to \mathbb{R}$ be a query , $\epsilon > 0$. Then the mechanism $\mathcal{M}_q : \mathsf{D} \times \Omega \to \mathbb{R}$ defined by $\mathcal{M}_q(d) = q(d) + w$, with $w \sim Lap(b)$, where $b \geq \frac{\Delta q}{\epsilon}$, is ϵ-differentially private [55].*

1.4.3 Pain-Free Algorithm

A persistent requirement in all DP tools is the need to bound global sensitivity, and in many applications, from collaborative filtering [126], and Bayesian inference [45] to anomaly detection [124], the principal challenge in ensuring differential privacy is to bound the sensitivity. Rubinstein et al. (the Pain-Free solution) [152] develop a simple approach to approximating global sensitivity with high probability. As shown in the following theorem, combined with generic mechanisms like Laplace, such a sampler enables a systematic realization of privacy protection.

Theorem 1.4.2. *Consider any database D of n records, privacy parameters $\epsilon > 0$, $\gamma \in (0, 1)$, sampling parameters size $m \in \mathbb{N}$, order statistic index $m \geq k \in \mathbb{N}$, approximation confidence $0 < \rho < min\{\gamma, 1/2\}$, and the known distribution P on D. The Pain-Free algorithm; which samples m sensitivity candidates, sorts them, and picks the kth one to calibrate the Laplace mechanism; preserves (ϵ, γ)-random differential privacy, where $m = \left\lceil \frac{\log(1/\rho)}{2(\gamma - \rho)^2} \right\rceil$, $k = \left\lceil m\big(1 - \gamma + \rho + \sqrt{\frac{\log(1/\rho)}{2m}} \right\rceil$, and $\rho = exp(W_{-1}(-\frac{\gamma}{2\sqrt{e}}) + 0.5)$.*

The derived specific expressions involve branches of the Lambert-W function, which is the inverse relation of the function $f(z) = z \cdot exp(z)$. Moreover, a number of natural choices for sampling distribution P could be made. In particular, the Pain-Free algorithm and its privacy guarantee are derived by assuming a *uniform* distribution defined over the domain of the dataset D. However,

the accuracy of the solution can be further boosted where a simulation process capable of approximating the actual P exists, e.g., the anomaly scores computed by an MSSP can be leveraged.

1.4.4 Notions and Notations

Database. We consider a database as a multiset of elements from a set $\mathcal{X}$, which is the set of possible values of records. In a database, we assume each record is associated with a distinct individual. We represent a database x as a histogram in $\mathcal{D} = \{y \in \mathbb{N}^{\mathcal{X}} : ||y||_1 < \infty\}$, where $\mathcal{D}$ is the set of all possible database, $\mathbb{N} = \{0, 1, 2, \cdots\}$, and x_i is the number of records in x that are identical to i.

Anomaly Identification Algorithm. The privacy definitions and constructions we develop are not tied to any specific anomaly definition. Specifically, DPOD relies on anomaly scores to define the distribution of the dataset as the underlying PDF of its sensitivity sampler. Therefore, any anomaly detection algorithm which assigns scores to the records to represent how outlying a record is [5, 4] can be successfully leveraged into its methodology. We note that comparison between the performance of such score-based anomaly detection algorithms is out of the scope of this work.

1.5 Utility Metrics

We review some background on utility metrics for the theoretical foundations of the R^{2}DP framework.

ℓ_p Metrics. In penalized regression, "ℓ_p penalty" refers to penalizing the ℓ_p norm of a solution's vector of parameter values (i.e., the sum of its absolute values, or its Euclidean length) [150]. In our privacy-utility setting, the ℓ_p utility metric is defined as follows.

Definition 1.5.1. *(ℓ_p). For a database mechanism $\mathcal{M}_q(D)$ the ℓ_p utility metric is defined as* $\mathbb{E}(|\mathcal{M}_q(D) - q(D)|^p)^{1/p}$.

Usefulness. Following Blum et al. [18], the following utility metric is commonly used for machine learning.

Definition 1.5.2. *(Usefulness). A mechanism $\mathcal{M}_q$ is (γ, ζ)-useful if, with probability $1-\zeta$, for any dataset $d \subseteq \mathsf{D}$, $|\mathcal{M}_q(d) - q(d)| \leq \gamma$.*

Theorem 1.5.1. *The Laplace Mechanism is $(\frac{\Delta q}{\epsilon} \ln \frac{1}{\zeta}, \zeta)$-useful, or equivalently, the Laplace Mechanism is $(\gamma, e^{\frac{-\gamma}{b(\epsilon)}})$-useful [33].*

Mallows Metric. The Mallows metric has been applied for evaluating the private estimation of the degree distribution of a social network [89]. It is defined to test if two samples are drawn from the same distribution. Given two random variables X and Y, we have $Mallows(X, Y) = \frac{1}{n}\sum_{i=1}^{n}(|X_i - Y_i|^p)^{1/p}$ (similar to p-norm).

Relative Entropy (Rényi Entropy). The relative entropy, also known as the *Kullback-Leibler (KL)* divergence, measures the distance between two probability distributions [39]. Formally, given two probability distributions $p(x)$ and $q(x)$ over a discrete random variable x, the relative entropy given by $D(p||q)$ is defined as follows: $D(p||q) = \sum_{x\in\mathcal{X}} p(x) \log \frac{p(x)}{q(x)}$. Further generalization came from Rényi [146, 77], who introduced an indexed family of generalized information and divergence measures akin to the Shannon entropy and KL divergence. Rényi introduced the entropy of order α as $I_\alpha(p||q) = \frac{1}{\alpha-1}\log(\sum_{x\in\mathcal{X}} p(x)^\alpha q(x)^{1-\alpha})$, $\alpha > 0$ and $\alpha \neq 1$.

Chapter 2

Literature Review

2.1 Network Trace Anonymization

In the context of anonymization of network traces, as surveyed in [131], many solutions have been proposed [153, 67, 135, 115, 127, 179, 83, 128, 44, 151]. In Figure 1, we have summarized the scope (e.g., accepted input data, anonymization fields, etc.) of some popular tools in anonymizing network traces. Generally, these may be classified into different categories, such as *enumeration* [64], *partitioning* [154], and *prefix-preserving* [172, 69]. These methods include removing rows or attributes (suppression) and generalization of rows or attributes [42]. Some of the solutions [147, 142] are designed to address specific attacks and are generally based on the permutation of some fields in the network trace to blur the adversary's knowledge. Later studies either prove theoretically [24] or validate empirically [29] that those works can be defeated by semantic attacks. There are only two tools that can resist semantic attacks, i.e., SCRUB [179] and TCPanon. Unfortunately, as shown in Figure 1, these two tools require heavy sanitization which render the released data less useful. We now review some important categories of network trace anonymization methods [44].

1. Format-preserving encryption [60]. This (pseudo-)random permutation of data can map original data to an encrypted version in the space of the original data. An example is the format

Tool name	Input Data Type										Anonymized Fields					Anonymization method					Weaknesses	
	Tcpdump	Netflow	Live interf	Nfdump	NCSA	Coralreef	PCAP	DAG	TSH	syslogs	Netflow fields	IP address	port	header	payload	Prefix-preserving	Permutation	Truncation	Precision Degradation	Enumeration	Highly sanitized	Semantic attacks
Anontool [1]	✓	✓	✓								✓	✓				✓	✓		✓		✓	✓
CANINE [3]		✓		✓	✓						✓	✓	✓			✓	✓			✓	✓	✓
CoralReef [2]	✓	✓				✓	✓	✓	✓		✓	✓	✓			✓	✓	✓				✓
Flaim [24]	✓	✓		✓						✓	✓	✓	✓			✓	✓		✓	✓	✓	✓
IPsumdump [21]	✓	✓					✓	✓	✓			✓		✓		✓						✓
NFDUMP [20]		✓	✓	✓							✓	✓			✓	✓					✓	✓
SCRUB [19]	✓	✓	✓									✓		✓	✓		✓				✓	
TCPanon [18]	✓						✓								✓		✓					
Tcpdpriv [17]	✓	✓					✓					✓		✓	✓	✓	✓	✓			✓	✓
Tcpmkpub [23]	✓						✓					✓		✓	✓	✓	✓	✓				✓
Tcpurify [22]	✓	✓					✓					✓			✓		✓	✓				✓

Figure 1: Network trace anonymization tools and semantic attacks

preserving encryption of credit-card numbers, which facilitates compatible input for existing devices yet provides some additional security against eavesdropping. Format-preserving encryption of IP or MAC addresses is extremely desirable in network trace anonymization as the anonymized version can be parsed by IDS or other network tools.

2. Prefix preservation [173]. Since prefixes in network traffic have special meanings, e.g., the first 6 bytes of a MAC address field (manufacturer) or the leading bytes of an IPv4 address (registered subnet), performing prefix preserving anonymization is highly desirable in that context. One big problem with this method is that the frequency is preserved due to the use of deterministic encryption, which can easily leak information about the true records since the attacker may know some prior background information of the frequency distribution. Specifically, Naveed et al. [136] present a series of attacks that recover the plaintext from deterministic encryption (DTE) and order-preserving encryption (OPE); using only the encrypted column and publicly-available auxiliary information. They have considered well-known attacks, including frequency analysis and sorting, as well as new attacks based on combinatorial optimization.

However, as we will show later in thus thesis, these attacks under multi-view can be successfully mitigated. We note that only the real view would preserve the true frequency of the records inside the dataset. Therefore, an adversary cannot apply existing frequency-based attacks as usual. Adversaries armed with some prior knowledge about frequencies can at most (as shown in Section 4.3.2) discard some of the fake views, and the remaining views will still be indistinguishable and render the frequency attacks difficult. To evaluate the impact of such an attack, our experiments have been based on the frequency analysis attack of Brekene et al. [13], which is a sophisticated attack for optimally leveraging the background knowledge to infer results using frequency analysis.

3. **Replacement [68].** This method applies a one-to-one mapping of a field to a new value of the same type. Moreover, to provide enough flexibility replacement is often applied with regular expressions and is suitable for both headers and full packet [103].

4. **Filtering and data removal [155].** Also called as truncation and black marking, results in data removal by overwriting it with fixed values, often zeros.

5. **Generalization [153].** This function is the act of replacing a data with more general data through partitioning (also called grouping or binning) information. For instance, TCP/UDP port numbers can be presented as ephemeral (≥ 1024) or non-ephemeral (< 1024).

6. **Precision degradation [64].** This method is comparable with black marking, whereas by degradation only the least significant information of a data field is removed. Examples include precision degradation of timestamps to less specific values. Another example is rounding of numeric values.

7. **Enumeration [153].** is an example of collision resistant mapping which provides order preservation and uniqueness. For example, applying enumeration to timestamps keeps the order but loses precision or distance.

8. Cryptographic permutation [24]. This function using a block cipher or hash function can be applied to uniquely permutate network data. The security of the cryptographic permutation depends largely on the input entropy. For instance, for very short values (e.g., 32 bit IPv4 addresses), a simple hashing is easily reversible due to lack of input entropy [23]. However, HMACs have better resistance to chosen plain text attacks than regular hashes [23].

Remarks. When choosing anonymization primitives, it is important to ensure compatibility, and, to some degree, meaningful transformations. Moreover, applying an anonyization method requires certain actions to be taken so that a valid network packet will be recieved (by IDS), e.g., TCP sequence/acknowledgement numbers (if used), all relevant checksums, and IP packet length fields should be corrected to reflect the proper data lengths [143].

As our proposed anonymization solution falls into the category of prefix-preserving solutions, which aims to improve the utility, we review in more details some of the proposed solutions in this category. First effort to find a prefix preserving anonymization was done by Greg Minshall [40] who developed TCPdpriv which is a table-based approach that generates a function randomly. Fan et al. [173] then developed CryptoPAn with a completely cryptographic approach. Several publications [23], [147, 142] have then raised the vulnerability of this scheme against semantic attacks which motivated query based [125] and bucketization based [148] solutions.

Among the works that address such semantic attacks, Riboni et al. [148] propose a (k,j)-obfuscation methodology applied to network traces. In this method, a flow is considered obfuscated if it cannot be linked, with greater assurance, to its (source and destination) IPs. First, network flows are divided into either confidential IP attributes or other fields that can be used to attack. Then, groups of k flows having similar fingerprints are first created, then bucketed, based on their fingerprints into groups of size $j < k$. However, utility remains a challenge in this solution, as the network flows are heavily sanitized, i.e., each flows is blurred inside a bucket of k flows having similar fingerprints. An alternative to the aforementioned solutions, called *mediated trace analysis* [132, 130], consists in performing the data analysis on the data-owner side and outsourcing

analysis reports to researchers requesting the analysis. In this case, data can only be analyzed where it is originally stored, which may not always be practical, and the outsourced report still needs to be sanitized prior to its outsourcing [125]. In contrast to those existing solutions, our approach improves both the privacy and utility at the cost of a slightly higher computational overhead.

2.2 The Optimal Mechanism in Differential Privacy

Differential privacy [55] is a model for preserving privacy while releasing the results of various useful functions, such as contingency tables, histograms and means [50]. Many existing works focus on improving the utility based on different mechanisms.

Noise Perturbation. Based on the general utility maximization framework from Ghosh et al. [76], Gupte and Sundararajan [82] further study the optimal noise probability distributions for single count queries. Later, Geng el al. [73, 72] demonstrate the optimal noise distribution has a Staircase-shaped PDF for Laplace mechanism. Furthermore, Balle and Wang [9] develop an optimal Gaussian mechanism in high privacy regime to minimize the noise and increase the utility for queries. Geng et al. [71] further show the optimal noise distribution is a uniform distribution over Gaussian mechanism. Moreover, Hardt et al. [87] study the privacy-utility trade-off for answering a set of linear queries over a histogram, where the error is defined as the worst expectation of the ℓ_2-norm (identical to variance) of the noise among all possible outputs. Subsequently, Brenner et al. [26] show that, for general query functions, no universally optimal DP mechanisms exist.

Sampling and Aggregation. Sampling and aggregation frameworks mostly split the database into chunks, and aggregate the result using a DP algorithm after querying each chunk [139]. To expand the applicability of output perturbation, Nissim et al. [139] propose a framework to formally analyze the effect of instance-based noise. Observing the highly compressible nature of many real-life data, researchers propose lossy compression techniques to add noise calibrated to the compressed data. Acs et al. [3] propose an optimization of Fourier perturbation algorithm that clusters and exploits the redundancy between bins. Instead of directly adding noise to histogram counts, it first

lossily compresses the data, then adds noise calibrated to the data. Li et al. [108] propose an algorithm to partition a data domain into uniform regions and adapt the strategy to fit the specific set of range queries to achieve a lower error rate. Zhang et al. [181] improve the clustering mechanism by sorting histogram bins based on the noisy counts.

Data Composition. Barak et al. [10] propose transforming the data into the Fouier domain, which could avoid the violation of consistency for low-order marginals in database tables. As efficiency is the main bottleneck for this approach when the number of attributes is large, Hay et al. [90] ensure that the error rate does not grow with the size of a database. The proposed hierarchical histogram method also achieves a lower error for a fixed domain. Different from one-dimensional datasets solution proposed by Hay et al. [90], Xiao et al. [170] propose *Privelet* that improves accuracy on datasets with arbitrary dimensions, which could reduce error to 25% compared to 70% as baseline error rate. Cormode et al. [41] apply *quadtrees* and *kd-trees* as new techniques for parameter setting to improve the accuracy on spatial data. Ding et al. [46] introduce a general noise-control framework on data cubes. Li et al. [109] unify the two range queries over histograms into one framework. Other techniques, such as principal component analysis (PCA), linear discriminant analysis (LDA) [95], and random projection [36, 171] are also used to lower the data dimension for reducing the errors. Cormode et al. [41] apply quadtrees (*data-independent*) and kd-trees (*data-dependent*) to add noise to a histogram output.

Adaptive Queries. In this technique, the improvement of utilities takes advantage of a known set of queries, for example, Dwork et al. [59] propose *Boosting for Queries* algorithm to obtain a better accuracy of learning algorithms. Hardt et al. [86, 85] present multiplicative weights mechanism to improve the efficiency of interactive queries. Instead of polynomial running time [57], this work achieves a nearly linear running time with a relaxed utility requirement. Yuan et al. [177, 178] propose low-rank mechanism (LRM) to further improve the adaptive queries. Other techniques such as correlated noise [138] and sparse vector technique (SVT) [120] are also used in adaptive queries.

Applications. Many researchers also work on improving the utility for different types of data, such as, the Fourier Perturbation Algorithm (FPA_k) [145] in time-series data (e.g., location traces, web history, and personal health), *kd-trees* on spatial data [41], and matrix-valued query [35].

Summary. Our R^2DP framework provides a complementary approach to those existing works by providing the opportunity of searching for the maximal utility along an extra dimension. This framework also enables data recipients to specify their utility requirements and the computed parameter could be incorporated into existing solutions to further improve utility.

2.3 Privacy Preserving Anomaly Detection

In this section, we review the most relevant works on privacy-preserving anomaly detection. We can distinguish two categories of works depending on whether they adopt differential privacy or not. In the following, we briefly review each category and compare them with our work.

Differentially Private Anomaly Detection. All works under this umbrella has recognized the paradox between anomaly detection and differential privacy pertaining to the indistinguishability property. Okada et al. [141] are the first to highlight such a conflict. To solve this paradox, a relaxation of the differential privacy is inevitable, particularly for existence-dependent queries, to ensure a certain level of detection accuracy. Thus, all reviewed DP-based works have defined their own new notion of the relaxed DP, which generally makes their respective approaches applicable to limited specialized cases (e.g., Anomaly-restricted privacy [141], and sensitive privacy [6]). In contrast, we borrow a well-known notion of relaxed DP, namely, the Random Differential Privacy (RDP) model [84]. Furthermore, most of the existing works do not consider outsourcing the anomaly detection task as they mainly rely on the fact that the anomaly detection is performed on the data in a private local settings and only the answers to the anomaly-related queries are provided in a deferentially private way. In the following, we review most relevant DP research for anomaly detection,

Okada et al. [141] introduce a mechanism based on the smooth upper bound of the local sensitivity (a relaxed DP) and use it in a limited setting. The approach does not report on the detected outliers but cover two types of deferentially private queries: outlier counting and top-h subspaces with large number of outliers.

Similarly, several other approaches (e.g., [17, 7, 8, 19]) propose to relax/generalize the notion of differential privacy and design methods to solve the anomaly detection task under this new definition. Most of the time, their approaches are data-dependent and can only be performed in a rather restricted setting. For instance, Bittner et al. [17] define the notion of anomaly-restricted differential privacy and propose a group-based search algorithm satisfying this relaxed notion. However, their approach relies on input dataset that have a single outlier, which is not a typical case in available datasets. Asif et al. [7, 8] consider differential privacy in the context of collaborative outlier detection where data is either vertically or horizontally distributed among multiple parties. However, the proposed approach works only for datasets with particular characteristics (i.e., categorical data) and its extension to numeric data using discretization can limit the types of detectable outliers, which harms the usability of the data. In contrast to those approaches, we consider a more general and practical model including any number of anomalies. Asif et al. [6] generalize the notion of differential privacy and define sensitive privacy, which determines sensitive records after quantifying the database, instead of assuming that being an outlier/inlier is independent of the database. The outlier model is considered data-dependent when a record is outlier only relative to the other records in the data. Böhler et al. [19] assume that the data owner already knows about the outliers and exclude them from the dataset before adapting the sensitivity to the rest of the data. Furthermore, their work rely on the distribution of trust between the analyst entity and a new entity called the correction server whose role is to increase the accuracy of the outlier detection results. Similarly, our approach rely also on anomaly exclusion, however, we do not assume that the data owner has any pre-knowledge about the anomalies and we consider complete outsourcing of the anomaly detection task to the analyst. We also note that exiting exclusion-based approaches cannot solve the challenges we are addressing without fundamentally being modified.

Privacy-Preserving without DP. While there are several approaches having the same objective of preserving privacy in anomaly detection, the privacy model they consider does not follow the state-of-the-art differential privacy one, which offers the best privacy protection as it is independent from any adversary background knowledge. In the following, for the sake of completeness, we provide a brief overview of these works.

Secure Multi-Party Computation. Several works ([160, 118, 174, 112, 30]) consider use of Secure Multiparty Computation (MPC) to preserve privacy while analysing the data. MPC relies on cryptographic algorithms to enable several participants to jointly compute specific functions over their private data without mutually sharing it with others. For instance, [160, 118] use MPC for privacy-preserving distance-based outlier detection, Xue et al. [174] propose privacy-preserving spatial outlier detection based on MPC and [112] for density-based outlier detection. However, MPC-based protocols are computationally expensive and implementing them practically still imposes several scalability and edfficiency challenges. To ease this issue, SEPIA [30] proposes an optimization of MPC comparison operations for processing high volume of network data in near real-time. It also designs privacy-preserving protocols for event correlation and aggregation of network traffic statistics (e.g., volume metrics addition, feature entropy computation, and distinct item count). However, SEPIA relies on several trusted servers, called privacy peers, which are responsible for all computations to be able to reconstruct the results. This imposes stringent trust and availability assumptions for the anomaly detection task. Furthermore, this computation model is not meant for outsourcing the anomaly detection task, which is one of our major objectives in this work.

Data Perturbation. Several solutions propose data perturbation techniques (e.g.,[38, 13, 61]) to preserve privacy while performing anomaly detection. While some works propose linear data perturbation approaches (e.g. [38]), it has been shown that those are generally prone to reverse engineering attacks under certain situations [117]. To address this issue, other works such as Bhaduri et al. [13] consider nonlinear random data perturbation. Other works (e.g., RMP [61]) propose to

combine both linear and non-linear perturbation. For instance, Erfani et al. [61] propose a privacy-preserving collaborative anomaly detection scheme called Random Multiparty Perturbation (RMP), which uses a combination of nonlinear and participant-specific linear perturbation. However, the privacy-preserving property of such schema was shown to be vulnerable to recovery attacks [119]. A limitation of such solutions is that their privacy need to be proven on a case by case basis.

Federated Learning. Several works leverage emerging learning models, namely federated learning, to enable several clients to collaborate on training a central ML model, while not sharing the training data. For instance, DÏoT [137] is the first to employ a federated learning approach to claim anomaly-detection while preserving privacy. A fundamental contrast to our work is that federated learning does not consider outsourcing the data for anomaly detection, but relies on training a local model based on the private data then sharing the updates to have a centralized model for anomaly detection. Furthermore, it has been demonstrated that simply maintaining data locality during training does not guarantee privacy preservation [159]. Truex et al. [159] propose a hybrid federated learning approach with differential privacy and MPC to balance accuracy and privacy. The approach is meant to mainly address the privacy issues of SMC-based techniques, however, the approach does not allow outsourcing anomaly detection.

Synthetic Data Generation from Original Data. Other works (e.g., [122]) rely on synthetic data generation to preserve the privacy of the original data. In this respect, new synthetic data is generated such that generated records are similar to the original ones while preserving the high-level relationships within the data, without actually disclosing real, single data points. Mayer et al. [122] study three state-of-the-art data synthesizers. While some of approaches generated synthetic data with as good utility as the original one, concerns about the exposure of such synthesized data to inference attacks require further investigations.

Chapter 3

A Multi-view Approach to Preserve Both Privacy and Utility in Network Trace Anonymization

3.1 Introduction

The owners of large-scale network data, ISPs and enterprises usually face a dilemma. As security monitoring and analytics grow more sophisticated, there is an increasing need for those organizations to outsource such tasks together with necessary network data to third-party analysts, e.g., Managed Security Service Providers (MSSPs) [47]. On the other hand, those organizations are typically reluctant to share their network trace data with third parties, mainly due to privacy concerns over sensitive information contained in such data. For example, important network configuration information, such as potential bottlenecks of the network, may be inferred from network traces and subsequently exploited by adversaries to increase the impact of the denial of service attacks [148].

In cases where data owners are convinced to share their network traces, the traces are typically subjected to some anonymization techniques. The anonymization of network traces has attracted significant attention (a more detailed review of related works will be given in Section 2.1).

For instance, *CryptoPAn* replaces real IP addresses inside network flows with prefix preserving pseudonyms, such that the hierarchical relationships among those addresses will be preserved to facilitate analyses [173]. Specifically, any two IP addresses sharing a prefix in the original trace will also do so in the anonymized trace. However, CryptoPAn is known to be vulnerable to the *fingerprinting* and *injection* attacks [23, 24, 176]. In those attacks, adversaries either already know some network flows in the original traces (by observing the network or from other relevant sources, e.g., DNS and WHOIS databases), or have deliberately injected some forged flows into such traces. By recognizing those known flows in the anonymized traces based on unchanged fields of the flows, namely, fingerprints (e.g., timestamps and protocols), the adversaries can extrapolate their knowledge to recognize other flows based on the shared prefixes [23]. We now demonstrate such an attack.

Example 3.1.1. *In Figure 2, the upper table shows the original trace, and the lower shows the trace anonymized using CryptoPAn. In this example, without loss of generality, we only focus on source IPs. Inside each table, similar prefixes are highlighted through similar shading.*

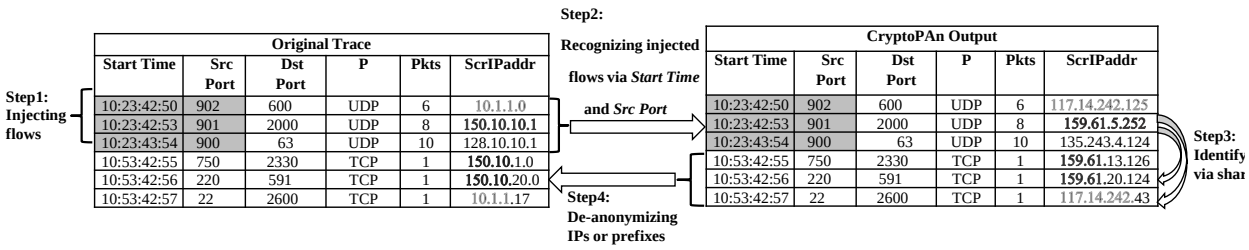


Figure 2: An example of injection attack

1. *Step 1: An adversary has injected 3 network flows, as the first 3 records in the original trace.*

2. *Step 2: The adversary recognizes the 3 injected flows in the anonymized trace (lower table) through unique combinations of the unchanged attributes (*Start Time *and* Src Port*).*

3. *Step 3: He/she can then extrapolate his/her knowledge from the injected flows to real flows as follows, e.g., since prefix* 159.61 *is shared by the second (injected), fifth (real) and sixth*

(real) flows, he/she knows all 3 must also share the same prefix in the original trace. Such identified relationships between flows in the two traces will be called matches *from now on.*

4. *Step 4: Finally, he/she can infer the prefixes or entire IPs of those anonymized flows in the original traces, as he/she knows the original IPs of his/her injected flows, e.g., the fifth and sixth flows must have prefix* 150.10*, and the IPs of the fourth and last flows must be* $10.1.1.0$*.*

A powerful adversary who probes all the subnets of a network via injection/fingerprinting can potentially de-anonymize the entire CryptoPAn output via a more sophisticated frequency analysis attack [23].

Most subsequent solutions either require heavy data sanitization or can only support limited types of analysis. In particular, the (k, j)-obfuscation method first groups together k or more flows with similar fingerprints and then bucketizes (i.e., replacing original IPs with identical IPs) $j < k$ flows inside each group; all records whose fingerprints are not sufficiently similar to $k - 1$ others will be suppressed [148]. Clearly, both the bucketization and suppression may lead to significant loss of data utility. The differentially private analysis method first adds noises to analysis results and then publishes such aggregated results [125]. Although this method may provide privacy guarantee regardless of adversarial knowledge, the perturbation and aggregation prevent its application to analyses that demand accurate or detailed records in the network traces.

In this work, we aim to preserve both privacy and utility by shifting the trade-off from between privacy and utility, as seen in most existing works, to between privacy and computational cost (which has seen a significant decrease lately, especially with the increasing popularity of cloud technology). The key idea is for the data owner to send enough information to the third party analysts such that they can generate and analyze many different anonymized views of the original network trace. Those anonymized views are designed to be sufficiently indistinguishable (which will be formally defined in Subsection 3.2.4) even to adversaries armed with prior knowledge and performing the aforementioned attacks, which preserves the privacy. On the other hand, one of the anonymized views will yield true analysis results, which will be privately retrieved by the

data owner or other authorized parties, which preserves the utility. More specifically, the major contributions are summarized as follows.

1. We propose a *multi-view* approach to the prefix-preserving anonymization of network traces. To the best of our knowledge, this is the first known solution that can achieve similar data utility as CryptoPAn does, while being robust against the so-called semantic attacks (e.g., fingerprinting and injection). In addition, we believe that the idea of shifting the trade-off from between privacy and utility to between privacy and computational cost can potentially be adapted to improve other privacy solutions.

2. In addition to the general multi-view approach, we detail a concrete solution based on iteratively applying CryptoPAn to each partition inside a network trace such that different partitions are anonymized differently in all the views except one (which yields valid analysis results that can be privately retrieved by the data owner). In addition to privacy and utility, we design the solution in such a way that only one *seed* view needs to be sent to the analysts, which avoids additional communication overhead.

3. We formally analyze the level of privacy guarantee achieved by our method, discuss potential attacks and solutions, and finally experimentally evaluate our solution using real network traces from a major ISP. The experimental results confirm that our solution is robust against semantic attacks with a reasonable computational cost.

3.2 Models

In this section, we describe models for the system and adversaries, we briefly review CryptoPAn, we provide a high level overview of our multi-view approach, and finally, we define our privacy property. Essential definitions and notations are summarized in Table 1.

3.2.1 The System and Adversary Model

Denote by $\mathcal{L}$ a *network trace* comprised of a set of *flows* (or records) r_i. Each flow includes a confidential multi-value attribute $A^{\text{IP}} = \{\text{IP}_{src}, \text{IP}_{dst}\}$, and the set of other attributes $A = \{A_i\}$ is called the *Fingerprint Quasi Identifier (fp-QI)* [148]. Suppose the data owner would like the analyst to perform an analysis on $\mathcal{L}$ to produce a report Γ. To ensure privacy, instead of sending $\mathcal{L}$, an anonymization function $\mathcal{T}$ is applied to obtain an anonymized version $\mathcal{L}^*$. Thus, our main objective is to find the anonymization function $\mathcal{T}$ to preserve both the *privacy*, which means the analyst cannot obtain $\mathcal{T}$ or $\mathcal{L}$ from $\mathcal{L}^*$, and *utility*, which means $\mathcal{T}$ must be prefix-preserving.

In this context, we make following assumptions (similar to those found in most existing works [173, 23, 24, 176]). i) The adversary is a honest-but-curious analyst (in the sense that he/she will exactly follow the approach) who can observe $\mathcal{L}^*$; ii) The anonymization function $\mathcal{T}$ is publicly known, but the corresponding anonymization key is not known by the adversary; iii) The goal of the adversary is to find all possible matches (as demonstrated in Example 3.1.1, an IP address may be matched to its anonymized version either through the fp-QI or shared prefixes) between $\mathcal{L}$ and $\mathcal{L}^*$; iv) Suppose $\mathcal{L}$ consists of d groups each of which contains IP addresses with similar prefixes (e.g., those in the same subset), and among these the adversary can successfully inject or fingerprint α ($\leq d$) groups (e.g., the demilitarized zone (DMZ) or other subnets to which the adversary has access). Accordingly, we say that the adversary has $\mathcal{S}_\alpha$ knowledge; v) Finally, we assume the communication between the data owner and the analyst is over a secure channel, and we do not consider integrity or availability issues (e.g., a malicious adversary may potentially alter or delete the analysis report).

Table 1: The Notation Table.

Symbol	Definition	Symbol	Definition
$\mathcal{L}$	Original network trace	$\mathcal{L}^*$	Anonymized trace
A^{IP}	IP attributes: source and destination IP	*fp-QI*	Fingerprint quasi identifier
r_i	Record number i	n	Number of records in $\mathcal{L}$
α	Number of IP prefixes known by the attacker	$\mathcal{S}_\alpha$	The set of addresses known by attacker
$\mathcal{S}_0^*$	The set of IP addresses in the seed view	$\mathcal{S}_i^*$	The set of IP addresses in view i
PP	CryptoPAn function	RPP	Reverse of CryptoPAn
P_i	Partition i	m	Number of partitions in $\mathcal{L}$
r	Index of real view	K_0, K_1	Private key and outsourced key

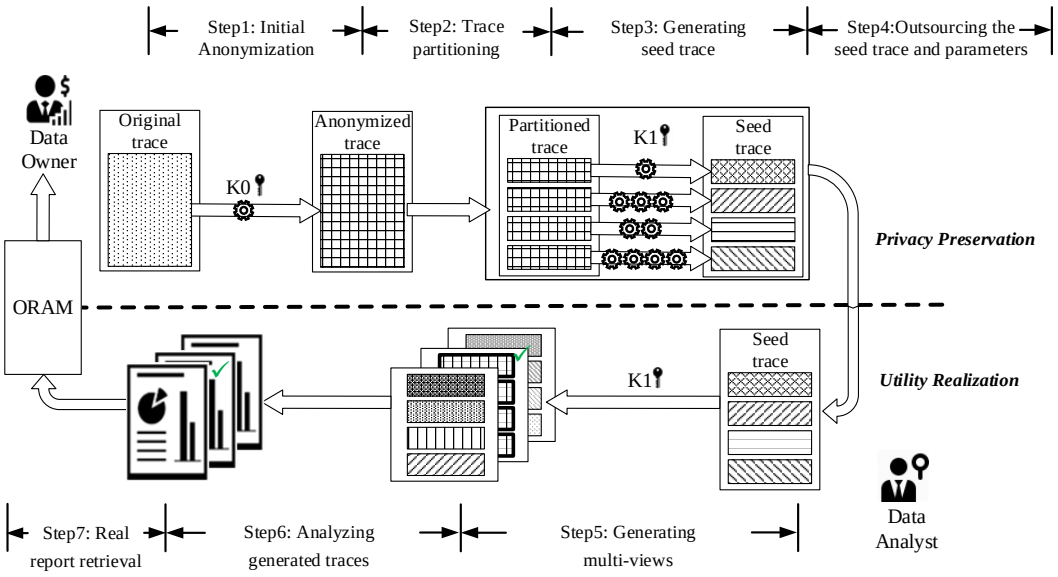


Figure 3: An overview of the multi-view approach

3.2.2 The CryptoPAn Model

To facilitate further discussions, we briefly review the CryptoPAn [173] model, which gives a baseline for prefix-preserving anonymization.

Definition 3.2.1. ***Prefix-preserving Anonymization*** *[173]: Given two IP addresses* $a = a_1a_2....a_{32}$ *and* $b = b_1b_2....b_{32}$*, and a one-to-one function* $F(.) : \{0,1\}^{32} \rightarrow \{0,1\}^{32}$*, we say that*

- a *and* b *share a* k*-bit prefix (*$0 \leq k \leq 32$*), if and only if* $a_1a_2....a_k = b_1b_2....b_k$*, and* $a_{k+1} \neq b_{k+1}$.

- F *is prefix-preserving, if, for any* a *and* b *that share a* k*-bit prefix,* $F(a)$ *and* $F(b)$ *also do so.*

Given $a = a_1a_2 \cdots a_{32}$ and $F(a) = a'_1a'_2 \ldots a'_{32}$, the prefix-preserving anonymization function F must necessarily satisfy the canonical form [173], as follows.

$$a'_i = a_i \oplus f_{i-1}(a_1a_2 \cdots a_{i-1}), \quad i = 1, 2, \ldots, 32 \tag{2}$$

where f_i is a cryptographic function which, based on a $256/128$-bit key K, takes as input a bit-string of length $i-1$ and returns a single bit. Intuitively, the i^{th} bit is anonymized based on K and

the preceding $i-1$ bits to satisfy the prefix-preserving property. The cryptographic function f_i can be constructed as $L\Big(R\big(P(a_1a_2\ldots a_{i-1}),K\big)\Big)$ where L returns the least significant bit, R can be a block cipher such as Rijndael [180], and P is a padding function that expands $a_1, a_2, \ldots, a_{i-1}$ to match the block size of R [173]. In the following, PP will stand for this CryptoPAn function and its output will be denoted by $a' = PP(a, K)$. The advantage of CryptoPAn is that it is deterministic and allows consistent prefix-preserving anonymization under the same K. However, as mentioned earlier, CryptoPAn is vulnerable to semantic attacks, which will be addressed in next section.

3.2.3 The Multi-View Approach

We propose a novel *multi-view* approach to the prefix-preserving anonymization of network traces. It preserves both the privacy and utility, while being robust against semantic attacks. The key idea is to hide a prefix-preserving anonymized view, namely, the *real view*, among $N-1$ other *fake views*, such that an adversary cannot distinguish between those N views, either using his/her prior knowledge or through semantic attacks. Our approach is depicted in Figure 3 and detailed below.

3.2.3.1 Privacy Preservation at the Data Owner Side

Step 1: The data owner generates two CryptoPAn keys K_0 and K_1, and then obtains an anonymized trace using the anonymization function PP (which will be represented by the gear icon inside this figure) and K_0. This initial anonymization step is designed to prevent the analyst from simulating the process as K_0 will never be given out. Note that this anonymized trace is still vulnerable to semantic attacks and must undergo the remaining steps. Besides, generating this anonymized trace will actually be slightly more complicated due to *migration* (see Section 3.3.3).

Step 2: The anonymized trace is then partitioned (as detailed in Sections 3.3.2 and 3.4).

Step 3: Each partition is anonymized using PP and key K_1, but the anonymization will be repeated, for a different number of times, on different partitions. For example, as the figure

shows, the first partition is anonymized only once, whereas the second for three times, etc. The result of this step is called the *seed trace*. The idea is that, as illustrated by the different graphic patterns inside the seed trace, different partitions have been anonymized differently, and hence the seed trace in its entirety is no longer prefix-preserving, even though each partition is still prefix-preserving (note that this is only a simplified demonstration of the *seed trace generator scheme* which will be detailed in Section 3.4).

Step 4: The seed trace and some additional parameters, including K_1, are outsourced to the analyst.

3.2.3.2 Utility Realization at the Data Analyst Side

Step 5: The analyst generates totally N *views* based on the received seed view and supplementary parameters. Our design will ensure one of those generated views, namely, the *real view*, will have all its partitions anonymized in the same way, and thus be prefix-preserving (detailed in Section 3.4), though the analyst (adversary) cannot tell which exactly is the real view.

Step 6: The analyst performs the analysis on all the N views and generates corresponding reports.

Step 7: The data owner retrieves the analysis report corresponding to the real view following an oblivious random access memory (ORAM) protocol [164], such that the analyst cannot learn which view has been retrieved.

Next, we define the privacy property for the multi-view solution.

3.2.4 Privacy Property against Adversaries

Under our multi-view approach, an analyst (adversary) will receive N different traces with identical fp-QI attribute values and different A^{IP} attribute values. Therefore, his/her goal now is to identify the real view among all the views, e.g., he/she may attempt to observe his/her injected or fingerprinted flows, or he/she can launch the aforementioned semantic attacks on those views, hoping that the real view might respond differently to those attacks. Therefore, the main objective

in designing an effective multi-view solution is to satisfy the *indistinguishability* property which means the real view must be sufficiently indistinguishable from the fake views under semantic attacks. Motivated by the concept of *Differential Privacy* [51], we propose the ϵ-indisinguishablity property as follows.

Definition 3.2.2. ***ϵ-Indistinguishable*** *Views: A multi-view solution is said to satisfy* ϵ-Indistingui-shability *against an $\mathcal{S}_\alpha$ adversary if and only if (from the adversary's point of view)*

$$\exists\, \epsilon \geq 0,\ \textit{s.t.}\ \forall i \in \{1, 2, \cdots, N\} \Rightarrow$$
$$e^{-\epsilon} \leq \frac{Pr(\textit{view } i \textit{ may be the real view })}{Pr(\textit{view } r \textit{ may be the real view})} \leq e^{\epsilon} \tag{3}$$

In Defintion 3.2.2, a smaller ϵ value is more desirable as it means the views are more indistinguishable from the real view to the adversary. For example, an extreme case of $\epsilon = 0$ would mean all the views are equally likely to be the real view to the adversary (from now on, we call these views the *real view candidates*). In practice, the value of ϵ would depend on the specific design of a multi-view solution and also on the adversary's prior knowledge, as will be detailed in the following sections.

Finally, since the multi-view approach requires outsourcing some supplementary parameters, we will also need to analyze the security/privacy of the communication protocol (privacy leakage in the protocol, which complements the privacy analysis in output of the protocol) in semi-honest model under the theory of secure multiparty computation (SMC) [175], [78] (see Section 3.4.3.2).

3.3 The Building Blocks

In this section, we introduce the building blocks for our approach, namely, the *iterative and reverse CryptoPAn*, *partition-based* prefix preserving, and *CryptoPAn with IP-collision (migration)*.

3.3.1 Iterative and Reverse CryptoPAn

As mentioned in Section 5.3.2, the multi-view approach relies on iteratively applying a prefix preserving function PP for generating the seed view. Also, the analyst will invert such an application of PP in order to obtain the real view (among fake views). Therefore, we first need to show how PP can be iteratively and reversely applied.

First, it is straightforward that PP can be iteratively applied, and the result also yields a valid prefix-preserving function. Specifically, denote by $PP^j(a, K)$ ($j > 1$) the *iterative* application of PP on IP address a using key K, where j is the number of iterations, called the *index*. For example, for an index of two, we have $PP^2(a, K) = PP(PP(a, K), K)$. It can be easily verified that given any two IP addresses a and b sharing a k-bit prefix, $PP^i(a, K)$ and $PP^i(b, K)$ will always result in two IP addresses that also share a k-bit prefix (i.e., PP^i is prefix-preserving). More generally, the same also holds for applying PP under a sequence of indices and keys (for both IPs), e.g., $PP^i(PP^j(a, K_0), K_1)$ and $PP^i(PP^j(b, K_0), K_1)$ will also share k-bit prefix. Finally, for a set of IP addresses $\mathcal{S}$, iterative PP using a single key K satisfies the following associative property $\forall \mathcal{S}, K,$ and $i, j \in \mathbb{Z}$ (integers): $PP^i\big(PP^j(\mathcal{S}, K), K\big) = PP^j\big(PP^i(\mathcal{S}, K), K\big) = PP^{(i+j)}\big(\mathcal{S}, K\big)$. On the other hand, when a negative number is used as the index, we have a *reverse* iterative CryptPAn function (RPP for short), as formally characterized in Theorem 3.3.1 (see proof in [133]).

Theorem 3.3.1. *Given IP addresses $a = a_1a_2 \cdots a_{32}$ and $b = PP(a, K) = b_1b_2 \cdots b_{32}$, the function $RPP(\cdot) : \{0, 1\}^{32} \rightarrow \{0, 1\}^{32}$ defined as $RPP(b, K) = c = c_1c_2 \cdots c_{32}$ where $c_i = b_i \oplus f_{i-1}(c_1 \cdots c_{i-1})$ is the inverse of the PP function given in Equation 2, i.e., $c = a$.*

3.3.2 Partition-based Prefix Preserving

As mentioned in Section 5.3.2, the core idea of the multi-view approach is to divide the trace into partitions (Step 2), and then anonymize those partitions iteratively, but for different number of iterations (Step 3). In this subsection, we discuss this concept.

Given $\mathcal{S}$ as a set of n IP addresses, we may divide $\mathcal{S}$ into partitions in various ways, e.g., forming

equal-sized partitions after sorting $\mathcal{S}$ based on either the IP addresses or corresponding timestamps. The partitioning scheme will have a major impact on the privacy, and we will discuss three such schemes in next section. Once the trace is divided into partitions, we can then apply PP on each partition separately, denoted by $PP(P_i, K)$ for the i^{th} partition. Specifically, given $\mathcal{S}$ divided as a set of m partitions $\{P_1, P_2, \cdots, P_m\}$, we define a *key vector* $V = \begin{bmatrix} v_1 & v_2 & \cdots & v_m \end{bmatrix}$ where each v_i is a positive integer indicating the number of times PP should be applied to P_i, namely, the *key index* of P_i. Given a cryptographic key K, we can then define the *partition-based* prefix preserving anonymization of $\mathcal{S}$ as $PP(\mathcal{S}, V, K) = \big[PP^{v_1}(P_1, K),\ PP^{v_2}(P_2, K), \ldots, PP^{v_m}(P_m, K)\big]$.

We can easily extend the associative property to this case as the following (which will play an important role in designing our multi-view mechanisms in next section).

$$PP[PP(\mathcal{S}, V_1, K), V_2, K] = PP(\mathcal{S}, (V_1 + V_2), K) \tag{4}$$

3.3.3 IP Migration: Introducing IP-Collision into CryptoPAn

As mentioned in Section 5.3.2, once the analyst (adversary) receives the seed view, he/she would generate many indistinguishable views among which only one, the real view, will be prefix preserving across all the partitions, while the other (fake) views do not preserve prefixes across the partitions (Step 5). However, the design would have a potential flaw under a direct application of CryptoPAn. Specifically, since the original CryptoPAn design is collision resistant [173], the fact that similar prefixes are only preserved in the real view across partitions would allow an adversary to easily distinguish the real view from others.

Example 3.3.1. *This vulnerability is shown in Figure 4. The initial trace includes three distinct addresses were split into two partitions P_1 and P_2 partitions. In the figure, the real view is easily distinguishable from the two fake views as the shared prefixes (159.61) between addresses in P_1 and P_2 only show up in the actual vision. This is because, since the partitions in fake views have different rounds of PP applied, and since the original CryptoPAn design is collision resistant [173],*

CryptoPAn (Collision Resistant)

3-View Defense			
Original Trace	**Fake View 1**	**Fake View 2**	**Real View**
P_1	$PP^4(P_1)$	$PP^2(P_1)$	$PP\ (P_1)$
150.10.10.1	144.5.116.249	50.19.13.26	159.61.5.252
128.10.10.1	39.250.139.225	83.180.10.3	135.243.4.124
P_2	$PP^3(P_2)$	$PP^5(P_2)$	$PP(P_2)$
150.10.20.0	17.8.78.28	159.61.20.124	159.61.20.124

Figure 4: An example showing only the real view contains shared prefixes (can be identified by adversaries)

the shared prefixes will no longer appear. Hence, the adversary can easily distinguish the real view from others.

To address this issue, our idea is to create collisions between different prefixes in fake views, such that adversaries cannot tell whether the shared prefixes are due to prefix preserving in the real view, or due to collisions in the fake views. However, due to the collision resistance property of CryptoPAn [173], there is only a negligible probability that different prefixes may become identical even after applying different iterations of PP, as shown in the above example. Therefore, our key idea of *IP migration* is to first replace the prefixes of all the IPs with common values (e.g., zeros), and then fabricate new prefixes for them by applying different iterations of PP. This IP migration process is designed to be prefix-preserving (i.e,. any IPs sharing prefixes in the original trace will still share the new prefixes), and to create collisions in fake views since the addition of key indices during view generation can easily collide. Next, we demonstrate this IP migration in an example.

Example 3.3.2. *In Figure 5, the primary stage shows the identical original trace as in Example 3.3.1. In the second stage, we "remove" the prefixes of all IPs and replace them with all zeros (by xoring them with their own prefixes). Next, in the third stage, we fabricate new prefixes by applying different iterations of PP in a prefix preserving manner, e.g., the first two IPs still sharing a*

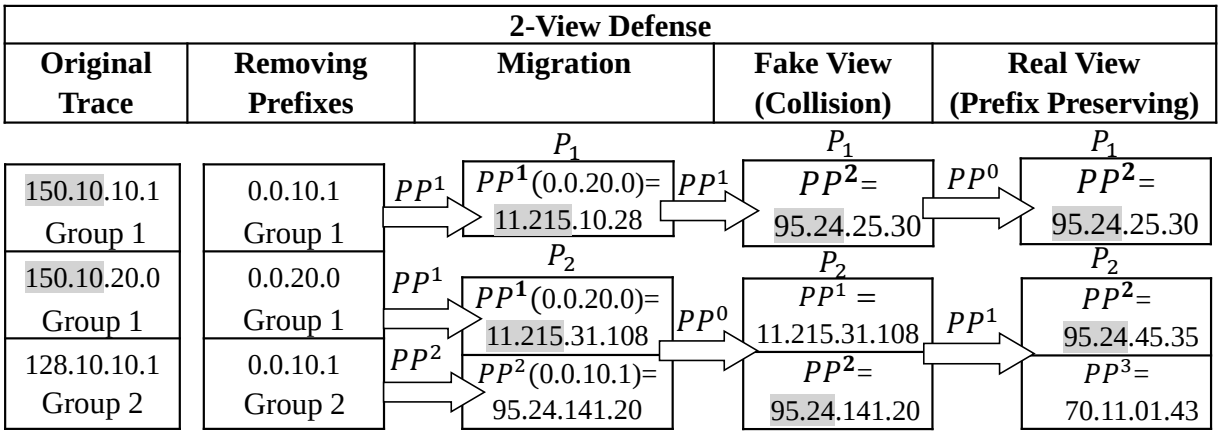


Figure 5: An example showing, by removing shared prefixes and fabricating them with the same rounds of PP, both fake view and real view may contain fake or real shared prefixes (which makes them indistinguishable)

common prefix (11.215*) different from that of the last IP. However, note that whether two IPs share the new prefixes only depends on their key indices now, e.g.,* 1 *for first two IPs and* 2 *for the last IP. This is how we can create collisions in the next stage (the fake view) where the first and last IPs coincidentally share the same prefix* 95.24 *due to their common key indices* 2 *(however, note these are the addition results of different key indices from the migration stage and the view generation stage, respectively). Now, the adversary will not be able to tell which of those views is real based on the existence of shared prefixes. We now formally define the migration function in the following.*

Definition 3.3.1. ***Migration Function****: Let* $\mathcal{S}$ *be a set of IP addresses consists of* d *groups of IPs* $S_1, S_2, \cdots, S_d$ *with distinct prefixes* $s_1, s_2, \cdots, s_d$ *respectively, and* K *be a random CryptoPAn key. Migration function* $M : \mathcal{S} \times \mathcal{C}$*(set of positive integers)* $\rightarrow \mathcal{S}^*$ *is defined as* $\mathcal{S}^* = M(\mathcal{S}) = \{S_i^* | \forall i \in \{1, 2, \cdots, d\}\}$ *where* $S_i^* = \{PP^{c_i}(s_i \oplus a_j, K), \forall a_j \in S_i\}$*, where* $\mathcal{C} = PRNG(d, d) = \{c_1, c_2, \cdots, c_d\}$ *is the set of* d *non-repeating random key indices generated between* $[1, d]$ *using a cryptographically secure pseudo random number generator.*

3.4 ϵ-Indistinguishable Multi-view Mechanisms

We first present a multi-view mechanism based on IP partitioning in Section 3.4.1. We then propose more refined schemes based on distinct IP partitioning with key vector generator in Section 3.4.2. Finally, we present a third scheme using random IP addresses permutation which can significantly reduce the cost in the data analyst side.

3.4.1 Scheme I: IP-based Partitioning Approach

To realize the main ideas of multi-view anonymization, as introduced in Section 5.3.2, we need to design concrete schemes for each step in Figure 3. The key idea of our first scheme is the following. We divide the original trace in such a way that all the IPs sharing prefixes will always be placed in the same partition. This will prevent the attack described in Section 3.3.3, i.e., identifying the real view by observing shared prefixes across different partitions. As we will detail in Section 3.4.1.4, this scheme can achieve perfect indistinguishability without the need for IP migration (introduced in Section 3.3.3), although it has its limitations which will be addressed in our second scheme. Both schemes are depicted in Figure 6 and detailed below. Specifically, our first scheme includes three main steps: privacy preservation (Section 3.4.1.1), utility realization (Section 3.4.1.2), and analysis report extraction (Section 3.4.1.3).

3.4.1.1 Privacy Preservation (Data Owner)

The data owner performs a set of actions to generate the seed trace $\mathcal{L}_0^*$ together with some parameters to be sent to the analyst for generating different views. These actions are detailed as follows.

- *Applying CryptoPAn using K_0:* First, the data owner generates two independent keys, namely K_0 (key used for initial anonymization, which never leaves the data owner) and K (key used for multi views generation step). The data owner then generates the initially anonymized trace $\mathcal{L}_0$=$PP(\mathcal{L}, K_0)$. This step is designed to prevent the adversary from simulating the

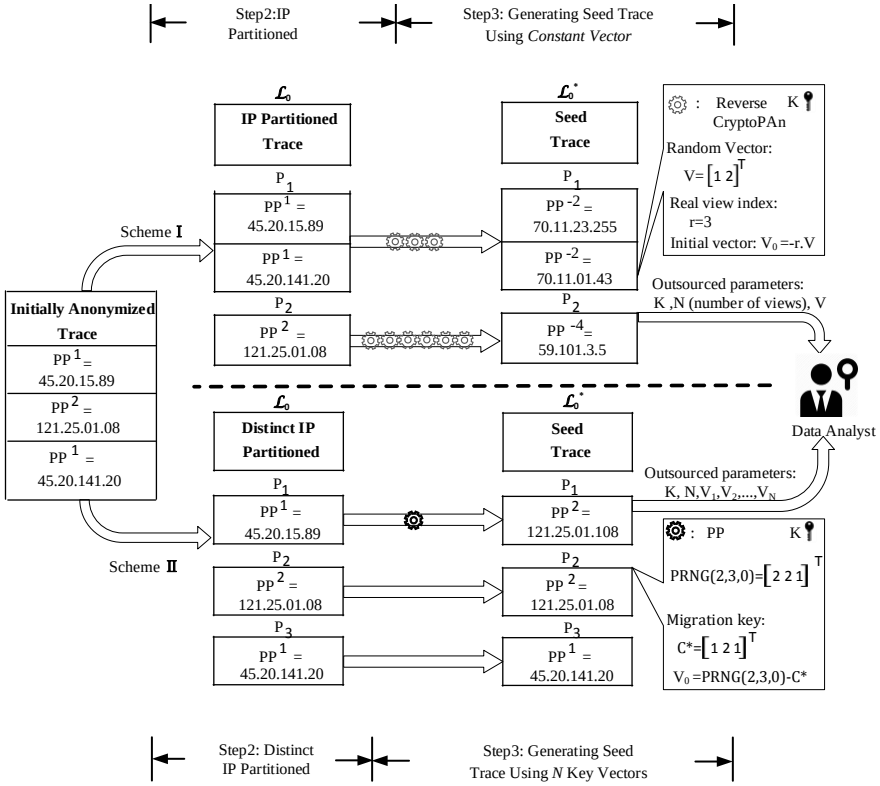


Figure 6: An example of a trace which undergoes multi-view schemes I, II

scheme, e.g., using a brute-force attack to revert the seed trace back to the original trace in which he/she can recognize some original IPs. The leftmost block in Figure 6 shows an example of the initially anonymized trace.

- *Trace partitioning based on IP-value:* The initially anonymized trace is partitioned based on IP values. Specifically, let $\mathcal{S}$ be the set of IP addresses in $\mathcal{L}_0$ consisting of d groups of IPs $S_1, S_2, \cdots, S_d$ with distinct prefixes $s_1, s_2, \cdots, s_d$, respectively; we divide $\mathcal{L}_0$ to d partitions, each of which is the collection of all records containing one of these groups. For example, the upper part of Figure 6 depicts our scheme I. The set of three IPs are divided into two partitions where P_1 includes both IPs sharing the same prefix, $45.20.15.89$ and $45.20.141.20$, whereas the last IP $121.25.01.08$ goes to P_2 since it does not share a prefix with others.

- *Seed trace creation:* The data owner in this step generates the seed trace using a d-size (recall that d is the number of partitions) random key vector.

– *Generating a random key vector:* The data owner generates a random vector V of size d using a cryptographically secure pseudo random number generator $PRNG(d\ , d)$ (which generates a set of d non-repeating random numbers between $[1, d]$). This vector V and the key K will later be used by the analyst to generate different views from the seed trace. For example, in Figure 6, for the two partitions, $V = \begin{bmatrix} 1 & 2 \end{bmatrix}$ is generated. Finally, the data owner chooses the total number of views N to be generated later by the analyst, based on his/her requirements about privacy and computational overhead, since a larger N will mean more computation by both the data owner and analyst but also more privacy (more real view candidates will be generated, and we will further study this via experiments).

– *Generating a seed trace key vector and a seed trace:* The data owner picks a random number $r \in [1, N]$ and then computes $V_0 = -r \cdot V$ as the key vector of seed trace. Next, the data owner generates the seed trace as $\mathcal{L}_0^* = PP(\mathcal{L}_0, V_0, K)$. This ensures, after the analysts applies exactly r iterations of V on the seed trace, he/she would get $\mathcal{L}_0$ back (while not being aware of this fact since he/she does not know r). For example, in Figure 6, $r = 3$ and $V_0 = \begin{bmatrix} -3 & -6 \end{bmatrix}$. We can easily verify that, if the analyst applies the indices in V on the seed trace three times, the outcome will be exactly $\mathcal{L}_0$ (the real view). This can be more formally stated as follows (the r^{th} view $\mathcal{L}_r^*$ is actually the real view). $\mathcal{L}_r^* = PP(\mathcal{L}_0^*, r \cdot V, K)$, using (4) which is to say $PP(\mathcal{L}_0, V_0 + r \cdot V, K)$, using (4), or identically $PP(\mathcal{L}_0, -r \cdot V + r \cdot V, K) = \mathcal{L}_0$.

- Outsourcing: Finally, the data owner outsources $\mathcal{L}_0^*$, V, N and K to the analyst.

3.4.1.2 Network Trace Analysis (Analyst)

The analyst generates the N views requested by the data owner, which is formalized as the following $\mathcal{L}_0^*$, is the seed view, thus $\mathcal{L}_i^* = PP(\mathcal{L}_{i-1}^*, V, K), \quad i \in \{1, \ldots, N\}$. Since boundaries of partitions must be recognizable by the analyst to allow him/her to generate the views, we modify

the timestamp of the records that are on the boundaries of each partition by changing the most significant digit of the time stamps which is easy to verify and does not affect the analysis as it can be reverted back to its original format by the analyst. Next, the analyst performs the requested analysis on all N views and generates N analysis reports $\Gamma_1, \Gamma_2, \cdots, \Gamma_N$.

3.4.1.3 Analysis Report Extraction (Data Owner)

The data owner is only interested in the analysis report that is related to the real view, which we denote by Γ_r. To minimize the communication overhead, instead of requesting all the analysis reports Γ_i of the generated views, the data owner can fetch only the one that is related to the real view Γ_r. He/she can employ the *oblivious random accesses memory* (ORAM) [164] to do so without revealing the information to the analyst (we will discuss alternatives in Section 2.1).

3.4.1.4 Security Analysis

We now analyze the level of indistinguishability provided by the scheme. The statement inside the probability in Definition 3.2.2 is the adversary's decision on a view, declaring it as fake or a *real view candidate*, using his/her $\mathcal{S}_\alpha$ knowledge. Moreover, we note that generated views differ only in their IP values (fp-QI attributes are similar for all the views). Hence, the adversary's decision can only be based on the published set of IPs in each view through comparing shared prefixes among those IP addresses which he/she already know ($\mathcal{S}_\alpha$). Accordingly, in the following, we define a function to represent all the prefix relations for a set of IPs.

Lemma 3.4.1. *For two IP addresses a and b, function $Q : \{0,1\}^{32} \times \{0,1\}^{32} \rightarrow \mathbb{N}$ returns the number of bits in the prefix shared between a and b: $Q(a,b) = 31 - \lfloor log_2^{a \oplus b} \rfloor$, where $\lfloor \cdot \rfloor$ denotes the* floor.

Definition 3.4.1. *For a multiset of n IP addresses $\mathcal{S}$, the* Prefixes Indicator Set *(PIS) $\mathcal{R}(\mathcal{S})$ is defined as follows: $\mathcal{R}(\mathcal{S}) = \{Q(a_i, a_j) | \, \forall a_i, a_j \in \mathcal{S}, i, j \in \{1, 2, \cdots, n\}\}$.*

Note that PIS remains unchanged when CryptoPAn is applied on $\mathcal{S}$, i.e., $\mathcal{R}(PP(\mathcal{S}, K)) = \mathcal{R}(\mathcal{S})$. In addition, since the multi-view solution keeps all the other attributes intact, the adversary can identify his/her pre-knowledge in each view and construct prefixes indicator sets out of them. Accordingly, we denote by $\mathcal{R}_{\alpha,i}$ the PIS constructed by the adversary in view i.

Definition 3.4.2. *Let $\mathcal{R}_\alpha$ be the PIS for the adversary's knowledge, and $\mathcal{R}_{\alpha,i}$, $i \in \{1, \cdots, N\}$ be the PIS constructed by the adversary in view i. A multi-view solution then generates* ϵ-indistinguishable *views against an $\mathcal{S}_\alpha$ adversary if and only if $\exists\ \epsilon \geq 0$, s.t. $\forall i \in \{1, 2, \cdots, N\} \Rightarrow e^{-\epsilon} \leq \frac{Pr(\mathcal{R}_{\alpha,i}=\mathcal{R}_\alpha)}{Pr(\mathcal{R}_{\alpha,r}=\mathcal{R}_\alpha)} \leq e^{\epsilon}$.*

Lemma 3.4.2. *The indistinguishability property, defined in Definition 3.4.2 can be simplified to $\exists\, \epsilon \geq 0,$ s.t. $\forall i \in \{1, 2, \cdots, N\} \Rightarrow Pr(\mathcal{R}_{\alpha,i} = \mathcal{R}_\alpha) \geq e^{-\epsilon}$.*

Proof. $Pr(\mathcal{R}_{\alpha,r} = \mathcal{R}_\alpha) = 1$ as view r is the prefix preserving output. Moreover, $\forall \epsilon \geq 0$, we have $e^\epsilon \geq 1$. Thus, we only need to show $\mathcal{R}_{\alpha,i} = \mathcal{R}_\alpha$ (each generated view i is a real view candidate). □

Theorem 3.4.3 (Proof in [133])**.** *Scheme* I *satisfies Definition 3.4.2 with $\epsilon = 0$.*

It shows that scheme I produces perfectly indistinguishable views ($\epsilon = 0$). In fact, it is robust against the attack explained in Section 3.3.3 and thus does not required IP migration, because the partitioning algorithm already prevents addresses with similar prefixes from going into different partitions (the case in Figure 4). However, although adversaries cannot identify the real view, they may choose to live with this fact, and attack each partition inside any (fake or real) view instead, using the same semantic attack as shown in Figure 2. Note that our multi-view approach is only designed to prevent attacks across different partitions, and each partition itself is essentially still the output of CryptoPAn and thus still inherits its weakness.

Fortunately, the multi-view approach gives us more flexibility in designing schemes to further mitigate such a weakness of CryptoPAn. We next present scheme II which sacrifices some indistinguishability (with slightly less real view candidates) to achieve better protected partitions.

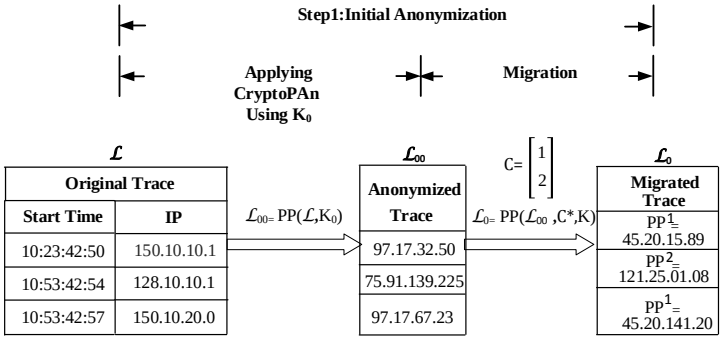


Figure 7: The updated initial anonymization (Step 1 in Figure 3) for enforcing migration

3.4.2 Scheme II: Multi-view Using N Key Vectors

To address the limitation of our first scheme, we propose the next scheme, which is different in terms of the initial anonymization step, IP partitioning, and key vectors for view generation.

3.4.2.1 Initial Anonymization with Migration

First, to mitigate the attack on each partition, we must relax the requirement that all shared prefixes go into the same partition. However, as soon as we do so, the attack of identifying the real view through prefixes shared across partitions, as demonstrated in Section 3.3.3, might become possible. Therefore, we modify the first step of the multi-view approach (initial anonymization) to enforce the IP migration technique. Figure 7 demonstrates this. The original trace is first anonymized with K_0, and then the anonymized trace goes through the migration process, which replaces the two different prefixes (97.17 and 75.91) with different iterations of PP, as discussed in Section 3.3.3.

3.4.2.2 Distinct IP Partitioning and N Key Vectors Generation

For the scheme, we employ a special case of IP partitioning where each partition includes exactly one distinct IP (i.e., the collection of all records containing the same IP). For example, the trace shown in Figure 6 includes three distinct IP addresses $150.10.10.1$,$128.10.10.1$, and $150.10.20.0$. Therefore, the trace is divided into three partitions. Next, the data owner will generate the seed view as in the first scheme, although the key V_0 will be generated completely differently, as detailed

below.

Let $\mathcal{S}^* = \{S_1^*, S_2^*, \cdots, S_d^*\}$, be the set of IP addresses after the migration step. Suppose $\mathcal{S}^*$ consists of D distinct IP addresses. We denote by $\mathcal{C}^*$ the multiset of totally D migration keys for those distinct IPs (in contrast, the number of migration keys in $\mathcal{C}$ is equal to the number of distinct prefixes, as discussed in Section 3.3.3). Also, let $PRNG(d, D, i)$ be the set of D random number generated between $[1, d]$ using a cryptographically secure pseudo random number generator at iteration i^{th}. The data owner will generate $N + 1$ key vector V_i as follows $V_i = PRNG(d, D, i) - PRNG(d, D, i-1), \forall i \neq r \in [1, 2 \cdots, N]$ and $V_0 = PRNG(d, D, 0) - \mathcal{C}^*$, $V_r = \mathcal{C}^* - PRNG(d, D, r-1)$.

Example 3.4.1. *In Figure 8, the migration and random vectors are* $\mathcal{C}^* = [1\ 1\ 2]$, $PRNG(2, 3, 0) = [1\ 2\ 2]$, $PRNG(2, 3, 1) = [1\ 2\ 1]$, *and* $PRNG(2, 3, 2) = [2\ 2\ 1]$, *respectively. The corresponding key vectors will be* $V_0 = [0\ 1\ 0]$, $V_1 = [0\ 0\ -1]$ *and* $V_2 = [1\ 0\ 0]$ *where only* V_1 *and* V_2 *are outsourced.*

In this scheme, the analyst at each iteration i generates a new set of IP addresses $\mathcal{S}_i^* = \{S_1^i, S_2^i, \cdots, S_d^i\}$ by randomly grouping all the distinct IP addresses into a set of d prefix groups. In doing so, each new vector V_i essentially cancels out the effect of the previous vector V_{i-1}, and thus introduces a new set of IP addresses $\mathcal{S}_i^*$ consisting of d prefix groups. Thus, we can verify that the r^{th} generated view will prefix preserving (the addresses are migrated back to their groups using $\mathcal{C}^*$).

Example 3.4.2. *Figure 8 shows that, in each iteration, a different set (but with an equal number of elements) of prefix groups will be generated. For example, in the seed view, IP addresses* 150.10.20.0 *and* 128.10.10.1 *are mapped to prefix group* 11.215.

3.4.2.3 Indistinguishability Analysis

By placing each distinct IP in a partition, our second scheme is not vulnerable to semantic attacks on each partition, since such a partition contains no information about the prefix relationship

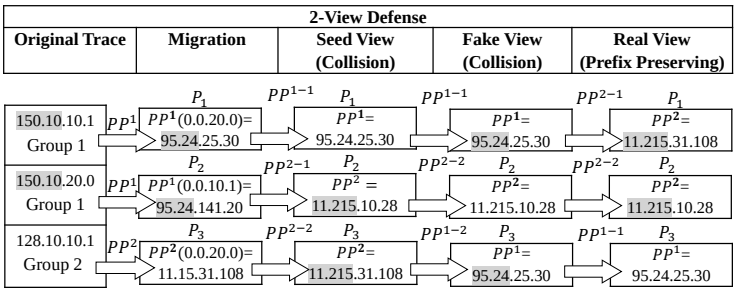


Figure 8: An example of two views generation under scheme II

among different addresses. However, compared with scheme I, as we show in the following, this scheme achieves a weaker level of indistinguishability (higher ϵ). Specifically, to verify the indistinguishability of the scheme, we calculate $Pr(\mathcal{R}_\alpha = \mathcal{R}_{\alpha,i})$ for scheme II in the following. First, the number of all possible outcomes of grouping D IP addresses into d groups with predefined cardinalities is $N_{total} = \frac{D!}{|S_1|!|S_2|!\cdots|S_d|!}$ where $|S_i|$ denotes the cardinality of group i. Also the number of all possible outcomes of grouping D IP addresses into d groups while still having $\mathcal{R}_{\alpha,i} = \mathcal{R}_\alpha$ is $N_{\text{real view candidates}} = \frac{\alpha!\ (D-\alpha)!\ \sum_{i=1}^{\binom{d}{\alpha}} \left(\Pi_{i=1}^{\alpha} |S_{a_i}|\right)}{|S_1|!|S_2|!\cdots|S_d|!}$ for some $a_i \in \{1, 2, \cdots, d\}$. This equation gives the number of outcomes when a specific set of α IP addresses ($\mathcal{S}_\alpha$) are distributed into α different groups and hence keeping $\mathcal{R}_{\alpha,i} = \mathcal{R}_\alpha$ (i.e., the adversary cannot identify collision). Note that term $\sum_{i=1}^{\binom{d}{\alpha}} \left(\Pi_{i=1}^{\alpha} |S_{a_i}|\right)$ is all the combinations of choosing this α groups for the numerator to model all the $(|S_{a_i}| - 1)!$ combinations.

Finally, we have $\forall i \leq N: \quad Pr(\mathcal{R}_{\alpha,i} = \mathcal{R}_\alpha) = \frac{N_{\text{real view candidates}}}{N_{total}}$ which is equal to

$$\mathcal{A} = \frac{\alpha!\ \sum_{i=1}^{\binom{d}{\alpha}} \left(\Pi_{i=1}^{\alpha} |S_{a_i}|\right)}{\Pi_{i=0}^{\alpha-1}(D-i)} \geq e^{-\epsilon} \tag{5}$$

Thus, to ensure the ϵ-indistinguishability, the data owner needs to satisfy the expression in equation 5 which is a relationship between the number of distinct IP addresses, the number of groups, the cardinality of the groups in the trace and the adversary's knowledge.

Theorem 3.4.4. *ϵ in the indistinguishability of the generated views in scheme* II *is lower-bounded*

by

$$\ln\big[\frac{D^\alpha}{d^\alpha}\cdot\Pi_{i=0}^{\alpha-1}\frac{(d-i)}{(D-i)}\big] \tag{6}$$

Proof. Let $b_1, b_2, \cdots, b_n$ be positive real numbers, and for $k = 1, 2, \cdots, n$ define the averages M_k as $M_k = \frac{\sum_{1\le i_1\le i_2\le\cdots\le i_k\le n} b_{i_1}b_{i_1}\cdots b_{i_k}}{\binom{n}{k}}$. By Maclaurin's inequality [15], which is the following chain of inequalities $M_1 \ge \sqrt[2]{M_2} \ge \sqrt[3]{M_3} \ge \cdots \ge \sqrt[n]{M_n}$ where $M_1 = \frac{\sum_{i=1}^{n} b_i}{n}$, we have $\mathcal{A} = \frac{\alpha!\binom{d}{\alpha}M_\alpha}{\Pi_{i=0}^{\alpha-1}(D-i)} \le \frac{\Pi_{i=0}^{\alpha-1}(d-i)(M_1)^\alpha}{\Pi_{i=0}^{\alpha-1}(D-i)}$ and since $M_1 = \frac{\sum_{i=1}^{n}|S_i|}{n} = \frac{D}{d}$, we have $A \le \frac{D^\alpha}{d^\alpha}\cdot\Pi_{i=0}^{\alpha-1}\frac{(d-i)}{(D-i)}$. □

Figure 9(a) shows how the lower-bound in Equation 6 changes with respect to different values of fraction d/D and also the adversary's knowledge. As it is expected, stronger adversaries have more power to weaken the scheme which results in increasing ϵ or increasing the chance of identifying the real view. Moreover, as it is illustrated in the figure, when fraction d/D grows, ϵ tends to converge to very small values. Hence, to decrease ϵ, the data owner may increase $d/D \in [0, 1]$ by grouping addresses based on a bigger number of bits in their prefixes, e.g., a certain combination of 3 octets would be considered as a prefix instead of one or two. Another solution could be aggregating the original trace with some other traces for which the cardinalities of each prefix group are small. We study this effect in our experiments in Section 3.6, especially in Figures 16 and 17.

Finally, Figure 9(b) shows how variance of the cardinalities affects the indistinguishability for a set of fixed parameters d, D, α. In fact, when the cardinalities of the prefix groups are close (small σ), $\mathcal{A}$ grows to meet the lower-bound in Theorem 3.4.4. Hence, from the data owner perspective, a trace with a lower variance of cardinalities and a bigger fraction d/D has a better chance of misleading adversaries who wants to identify the real view.

3.4.2.4 Security of the communication protocol

We now analyze the security/privacy of our protocol in semi-honest model under the secure multi-party computation (SMC) [175], [78] theory.

Lemma 3.4.5 (Proof in [133])**.** *Scheme* II *only reveals the CryptoPAn Key* K *and the seed trace*

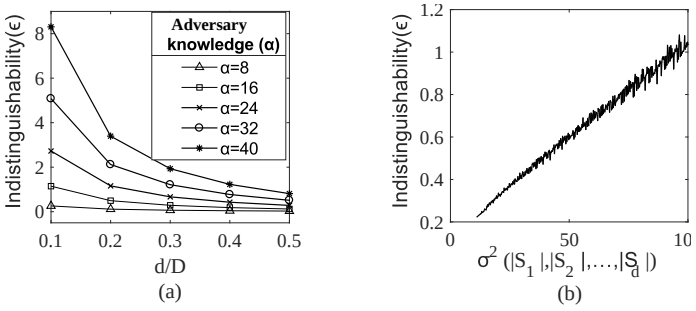


Figure 9: (a) The trend of bound 6 for ϵ when adversary's knowledge varies. (b) The trend of exact value of ϵ in equation 5 for $\alpha = 16$, $d/D = 0.1$ and when variance of cardinalities varies

$\mathcal{L}_0^*$.

Note that outsourcing the $\mathcal{L}_0^*$ and the outsourced key are trivial leakage. The outsourced key can be considered as a public key and the leakage of $\mathcal{L}_0^*$ which was studied earlier. Finally, we study the *setup leakage* and show that the adversary cannot exploit outsourced parameters to increase ϵ (i.e., decrease the number of real view candidates) by building his/her own key vector.

Lemma 3.4.6 (Proof in [133])**.** *For an $\mathcal{S}_\alpha$ adversary, who wants to obtain the least number of real view candidates, if condition $(2d-2)^D > N$ holds, the best approach is to follow scheme* II*, (scheme* II *returns the least number of real view candidates).*

3.4.3 Scheme III: IP Attribute Permutation

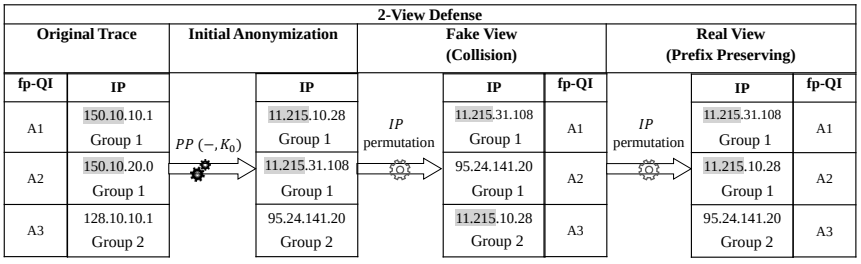


Figure 10: An example of two views generation under scheme III

As we have seen, scheme II requires applying CryptoPAn over the (sheer size) network trace for a large number of times. Therefore, we propose a third scheme which achieves the same indistinguishability bound as scheme II does among the generated views, yet requires much less computation cost in its deployment. The key idea is to fulfill the IP distribution task required in each view

generation by randomly permuting/assigning IP addresses to other attributes. Specifically, instead of using associative property of CryptoPAn to randomly diffuse IP addresses in the fake views and reconstruct the real view, in scheme III, we simply permute IP addresses and randomly assign to them the fp-QI attributes to achieve the diffusion and the reconstruction. The following example demonstrates different operations to be performed on a sample network trace under scheme III.

Example 3.4.3. *Figure 10 is an example of generating two views under scheme* III*. The original trace is illustrated in Figure 10 with three records and each record is a pair (fp-QI Activity, IP). As illustrated in Figure 10, the CryptoPAn is applied only once in the initialization step on the data owner's side. The view generation is then performed through shuffling (or permuting) the IP addresses and assigning them to one of the fp-QI activities. For example, in Figure 10, the fake view is generated by switching the IP addresses of group 1 with those from group 2, and the real view is reconstructed by permuting IP addresses back to their original positions. Clearly, the IP permutation algorithm is the main building block of this scheme. To satisfy the same level of indistinguishability as scheme* II*, the permutation algorithm follows the key vector generation method of scheme* III*. The following steps are the main operations that the data owner needs to perform under scheme* III*.*

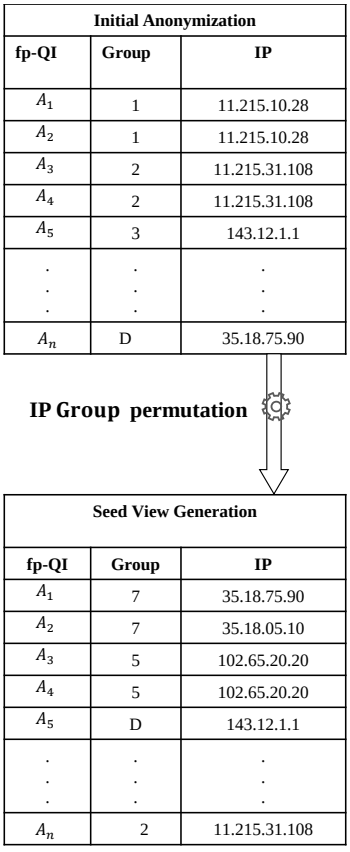


Figure 11: Seed view generation in scheme III for a trace of n records and D distinct IPs

Input:

$\mathcal{L}$, $\mathcal{S}$, K_0, r, D, fp-QI: same as in scheme II

$V_0, V_1, \cdots, V_N$: Vectors of size D defined by data owner same as in scheme II

Permute $(\mathcal{S}, V)$: Permuting Distinct IP Addresses

Assign $(fp-QI, \mathcal{S})$: Assigning IPs to Corresponding fp-QI

Output:

$\mathcal{L}^*$: Anonymized trace to be outsourced

Function: anonymize $(\mathcal{L}, D, K_0, r, V_0)$

begin

$\mathcal{L} := PP(\mathcal{L}, K_0)$

$\mathcal{S}^* :=$ Permute $(\mathcal{S}, V_0)$

$\mathcal{L}^* :=$ Assign $(fp-QI, \mathcal{S}^*)$

end

return $\mathcal{L}^*$, $V_1, V_2, \cdots, V_N$

end

Algorithm 1: Data owner: network trace anomymization (scheme III)

- *Initial anonymization*: The data owner generates the initially anonymized IP addresses using the set of original IPs $\mathcal{S}$ and a confidential key K_0, i.e., $\mathcal{S}^* = PP(\mathcal{S}, K_0)$.

- *N+1 permutation vectors generation*: The data owner then generates $N + 1$ permutation vectors to be used by the data analyst to perform permutation of IP addresses. Specifically, $V_i = PRNG(D, D, i)$, $i \in \{0, 1, \cdots, N\}$, is the set of D distinct random numbers generated between $[1, D]$ using a cryptographically secure pseudo random number generator at iteration i.

- *Seed trace creation*: After generating vector V_0, the data owner performs permutation on the initially anonymized trace for each distinct IP according to the group indices in random

vector V_0. Figure 11 illustrates the seed view generation under this scheme.

Algorithm 1 summarizes these actions. Finally, the operations of the data analyst in generating different views and analyze them are identical to the one under scheme II.

3.4.3.1 Indistinguishability Analysis

By placing each distinct IP in a partition through permutation, as we show in the following, our third scheme is not vulnerable to semantic attacks on each partition similar to scheme II. Specifically, to verify the indistinguishability of the scheme, we calculate $Pr(\mathcal{R}_\alpha = \mathcal{R}_{\alpha,i})$ for scheme II in the following. First, the number of all possible outcomes of permuting D is $N_{total} = \frac{D!}{|S_1|!|S_2|!\cdots|S_d|!}$, where $|S_i|$ denotes the cardinality of group i. Also the number of all possible outcomes of permuting D IP addresses while still having $\mathcal{R}_{\alpha,i} = \mathcal{R}_\alpha$ is $N_{\text{real view candidates}} = \frac{\alpha!\ (D-\alpha)!\ \sum_{i=1}^{\binom{d}{\alpha}} \left(\Pi_{i=1}^{\alpha}|S_{a_i}|\right)}{|S_1|!|S_2|!\cdots|S_d|!}$ for some $a_i \in \{1, 2, \cdots, d\}$ (refer to scheme II for details). Finally, we have $\forall i \leq N: \quad Pr(\mathcal{R}_{\alpha,i} = \mathcal{R}_\alpha) = \frac{N_{\text{real view candidates}}}{N_{total}} = \frac{\alpha!\ \sum_{i=1}^{\binom{d}{\alpha}} \left(\Pi_{i=1}^{\alpha}|S_{a_i}|\right)}{\Pi_{i=0}^{\alpha-1}(D-i)} \geq e^{-\epsilon}$. Thus, the ϵ-indistinguishability of scheme III is identical to the one in scheme II.

Corollary 3.4.1. *The indistinguishability parameter ϵ of the generated views in scheme* III *is lower-bounded by* $\ln\left[\frac{D^\alpha}{d^\alpha} \cdot \Pi_{i=0}^{\alpha-1} \frac{(d-i)}{(D-i)}\right]$.

Similar to scheme II, the more the fraction d/D is the better the indistinguishability of the scheme will be. Hence, to decrease ϵ, the data owner may increase $d/D \in [0, 1]$ by grouping addresses based on a bigger number of bits in their prefixes, e.g., a certain combination of 3 octets would be considered as a prefix instead of one or two.

3.4.3.2 Security of the communication protocol

We now analyze the security/privacy of our communication protocol in semi-honest model under the theory of secure multiparty computation (SMC) [175], [78].

Lemma 3.4.7. *Scheme* III *only reveals the seed trace $\mathcal{L}_0^*$ in semi-honest model.*

Proof. Since our communication protocol only involves one-round communication between two parties, we only need to examine the data analyst's view (messages received from the protocol), which includes (1) $\mathcal{L}_0^*$: the seed trace, and (2) $V_1, V_2, \cdots, V_N$: the key vectors. Similar to scheme II, each of $V_1, V_2, \ldots, V_N$ can be simulated by generating a single random number from a uniform random distribution (which proves that they are not leakage in the protocol). Specifically, all the entries in $V_1, V_2, \cdots, V_N$ are in $[1, D]$. Thus, all the random entrees in $V_1, V_2, \cdots, V_N$ can be simulated in polynomial time using a simulator (based on the knowledge data analyst already knew, i.e., his/her input and/or output of the protocol). □

Finally, similar to scheme II, we study the *setup leakage*.

Lemma 3.4.8. *For an $\mathcal{S}_\alpha$ adversary, who wants to obtain the least number of real view candidates, if condition $(2d-2)^D > N$ holds, the best approach is to follow scheme* III*, (scheme* III *returns the least number of real view candidates).*

3.5 Application of the Multi-view Approach in Other Domains

In this section, we discuss some important applications of the multi-view approach in other privacy-preserving contexts. In particular, we extend (1) our framework to leverage many other datasets, e.g., location data and genome data, and (2) the multi-view model to be used in designing non-interactive schemes guaranteeing differential privacy with boosted utility.

3.5.1 Multi-view for Outsourcing Other Types of Datasets

Multi-view represents a very general concept which could potentially be applied in a broader range of contexts. In fact, Property preserving encryption (PPE) techniques have significantly advanced the utility of encrypted data in various data outsourcing settings (e.g., the cloud) [116]. However, while preserving certain properties (e.g., order and prefixes) in the encrypted data, PPEs are typically limited to specific data types (e.g., IP addresses) [173], applications (e.g., range query) [116]

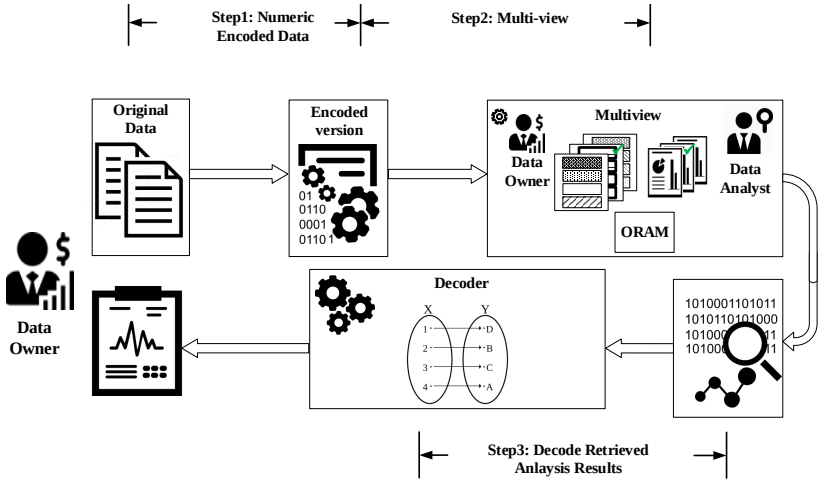


Figure 12: Multi-view can be applied to many other data types using data encoder and decoder patches

and are highly vulnerable to the aforementioned inference attacks which greatly limit their applications in practice. Therefore, we consider the potential of generalizing our multi-view approach to various data types (e.g., geo-locations, DNA sequences, market basket data, numeric data and timestamps) while providing an additional layer of protection against the inference attacks. For instance, location data includes the two-dimensional latitude and longitude coordinates of different places, which are highly precise float numbers (up to 8 decimal digits). In the *Bing Map Tiles System*,[1] the map is recursively divided into four tiles with equal size to reach the required resolution for quick map zoom in/out. Motivated by such a hierarchical data or structures, we can encode the coordinates into bit strings by concatenating the index of each level for one specific location , e.g., the coordinates of "New York" are $(40.730610, -73.935242)$. At the 23rd level, the pixel coordinates are $(632700926, 807272436)$ and tile coordinates are $(2471487, 3153407)$. Then, the encoding bit string can be derived as "e1147b6afff" in hexadecimal format.

In the encoded bit strings for coordinates, if prefixes can be preserved in the encrypted locations (while preserving the privacy), utility can be significantly preserved for analysis. For instance, "central park" and "the empire state building" in New York share a prefix, and the encrypted data

[2]https://docs.microsoft.com/en-us/bingmaps/articles/bing-maps-tile-system

for these two locations should also share the same length of prefix (e.g., two other places in London with the same level of proximity). Thus, the structure of the locations and the distance between such locations (besides other features such as frequency for property preserving encryption and deterministic encryption) can be preserved in the outsourced data. Next by feeding this numeric encoded version of the original data to our multi-view approach and subsequently decoding the numeric analysis results as shown in Figure 12, data owners can benefit from accurate analysis results while being protected from a variety of inference attacks on PPEs [48, 16, 81].

We note that the utility of the location data under the multi-view solution will only be partially preserved, depending on the common prefixes which can be determined by different partitioning schemes and the privacy protection level requested by the data owner. Like the network data, the prefixes are only preserved inside the same partition under the multi-view solution. Thus, the distances between locations can be preserved inside the same partition (with the common prefixes). Given larger partitions, more utility will be preserved with longer common prefixes in the output. Nonetheless, applying the multi-view solution with property preserving encryption can still be found useful in many applications as the semantic of the data is mostly preserved for different types of analyses.

3.5.2 Multi-view and Differential Privacy

Differential privacy (DP) has emerged as a de facto standard privacy notion for a wide range of applications [63, 14]. By requiring the presence of any individual's data in the input to only marginally affect the distribution over the output, differential privacy provides strong protection against adversaries with arbitrary background knowledge about the individuals. Differentially private mechanisms are typically special-purpose algorithms developed for specific applications, e.g., [63, 96, 14]. Many of those existing works address the interactive scenario, i.e., they provide perturbed answers to (limited sets of) queries. In contrast, micro-data publishing methods aim to release a sanitized dataset that supports a variety of use cases [183]. On the other hand, development

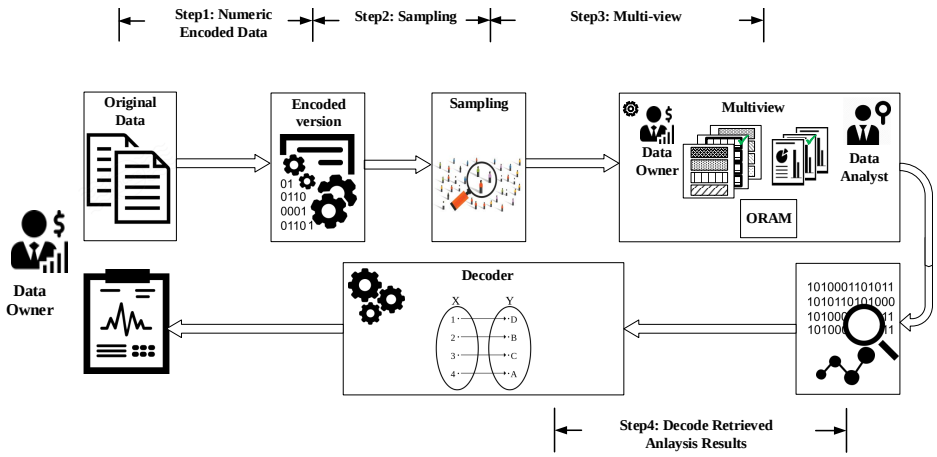


Figure 13: Multi-view can be remodeled (using sampling) to achieve differential privacy

of non-interactive methods which satisfy differential privacy while retaining sufficient data quality has remained challenging. Previous work has shown that algorithms which draw a random sample of data followed by generic k-anonymization can fulfill differential privacy [70, 113, 114]. These results are notable, as they combine statistical disclosure control, data anonymization and differential privacy. Specifically, Ninghui li et al. [46] show that any strongly-safe k-anonymization algorithm satisfies (like the multi-view approach) $(\beta, \epsilon, \delta)$-DPS (Differential Privacy under Sampling) for any $0 < \beta < 1, \epsilon \geq -\log(1-\beta)$, and $\delta = d(\beta, \epsilon, \delta)$, where d is a function of other parameters. Based upon these approaches, we consider the application of the multi-view approach for implementing a non-interactive scheme with differential privacy guarantee. Li et al. [46] showed that sampling records and reducing the uniqueness of their features through generic k-anonymity ensure Differential Privacy.

The multi-view approach can provide such a generic (independent from the dataset) k-anonymity solution through generating multiple views from one view. Specifically, since all attributes other than IPs are kept intact, and since the inter-partition prefix relation between IPs is not preserved, by properly grouping IP values (so that IP partitions become identical), multi-view can provide such generic k-anonymity, where k is the number of views. We note that here the number of views must always be in order of m! where m is the number of partitions after sampling. This is because the

multi-view solution to be a perfect k-anonymous algorithm must generate all possible instances of a k-anonymous data. Figure 13 illustrates different steps of generating a $(0.35, 8 \times 10^{-6})$-DP network trace out of an original trace with four records by means of 0.4-sampling and two view generation.

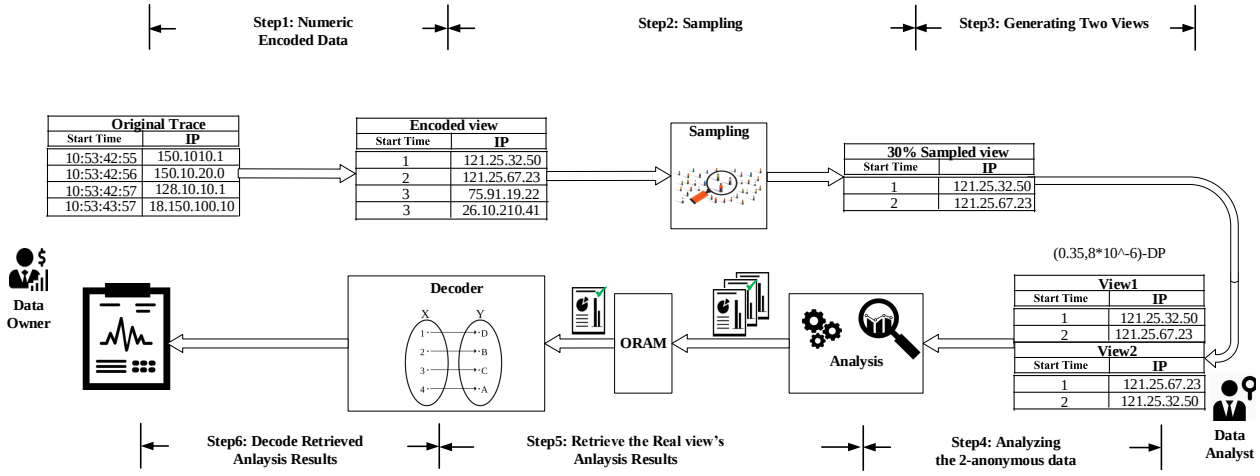


Figure 14: An example of differentially private network trace generation using multi-view

3.6 Experiments

This section evaluates our multi-view scheme through experiments with real data. I would like to thank Momen Oqaily for his help in conducting the following results.

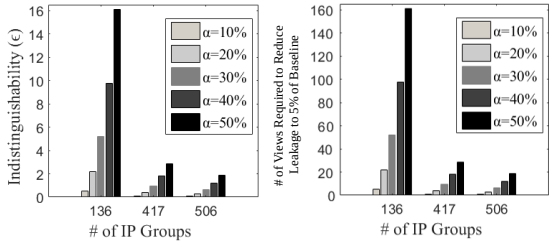
3.6.1 Setup

To validate our multi-view anonymization approach, we use a set of real world network traces collected by an anonymous ISP. We focus on attributes $Timestamp$, $IPaddress$, and $PacketSize$ in our experiments, and the meta-data are summarized in the table in Figure 15(a).

In order to measure the security of the proposed approach, we implement the frequency analysis

Attribute	Value
Type Header	Header
Start date	August 1st, 2016
End date	October 21st, 2016
Size	70 GB
Format	pcap and TXT
Distinct IP addresses	883
Number of packets	1M

(a)



(b) (c)

Figure 15: (a) Metadata of the collected traces (b) ϵ for different number of prefix groups and different adversary knowledge, and (c) the required number of views to reduce the leakage of CryptoPAn down to 5%

attack [136], [23]. This attack can compromise individual addresses protected by existing prefix-preserving anonymization in multi-linear time [23]. We note that in the setting of EDBs (encrypted database systems), an attack is successful if it recovers even partial information about a single cell of the DB [136]. Accordingly, we define the information leakage metric to evaluate the effectiveness of our solution against the adversary's semantic attacks. Several measures have been proposed in literature [173, 147] to evaluate the impact of semantic attacks. Motivated by [173], we model the information leakage (number of matches) as the number of records/packets, their original IP addresses are known by the adversary either fully or partially. More formally,

Information leakage metric [173]: We measure F_i defined as the total number of addresses that has at least i most significant bits known, where $i \in \{1, 2, \cdots, 32\}$.

To model adversarial knowledge, we define a set of prefixes to be known by the adversary ranging from 10% up to 100% of all the prefixes in the trace. This knowledge is stored in a two dimensional vector that includes α different addresses and their key indexes. Next, using our multi-view schemes, we generate all the N views. Before we apply the frequency analysis attack, we simulate how an adversary may eliminate some fake views from further consideration as follows. For each view, we check if two addresses from the adversary's knowledge set with different prefixes now share prefixes in that view. If we find such a match in the key indices, the corresponding view will be discarded from the set of the real view candidates and will not be considered in our

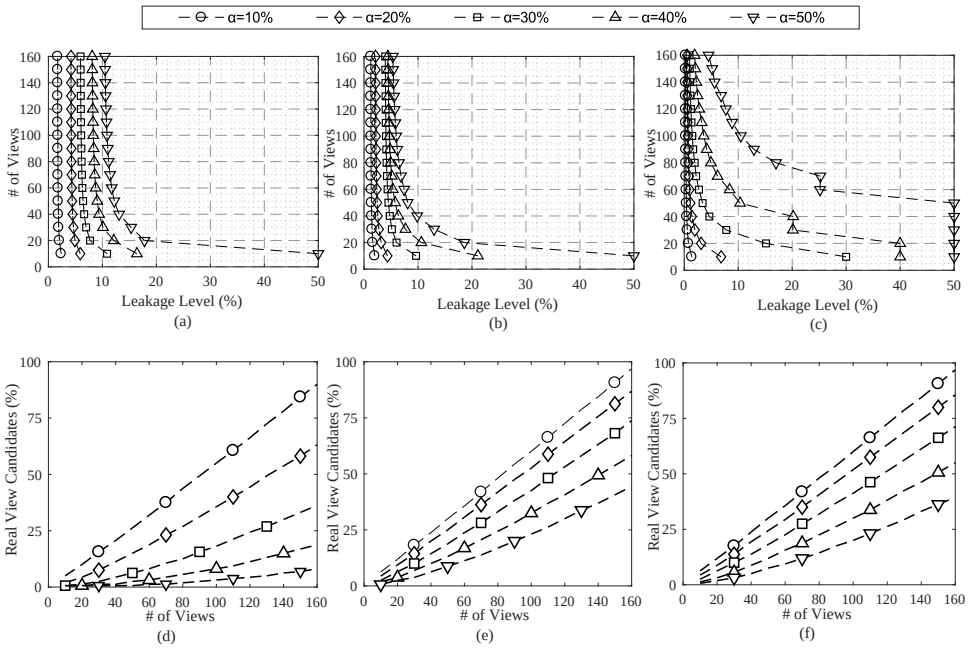


Figure 16: Percentage of the compromised packets in schemes II, III (out of 1M) and number of real view candidates when number of views and the adversary knowledge vary and for the three different cases (1) Figures (a),(d) (2) Figures (b),(e) (3) Figures (c),(f)

experiments since the adversary would know it is a fake view.

In the following, we present an extensive set of experiments evaluating the utility and the performance of the multi-view approach. Moreover, we justify the choice of ORAM in our setup using a comprehensive study on the scalability of ORAM in the literature. We validate the effectiveness of our schemes by showing the number of real view candidates and the percentage of the packets in the trace that are compromised (i.e., the percentage of IP packets whose addresses have at least eight most significant bits known). Each experiment is repeated more than $1,000$ times and the end results are the average results of the frequency analysis algorithm applied to each of the real view candidates. Finally, we evaluate the performance of our solution by measuring memory and CPU consumption as well as the time required to run our solution for one million packets while increasing the number of generated views. We conduct all experiments on a machine running Windows with an Intel(R) Core(TM) i7-6700 3.40 GHz CPU, 4 GB Memory, and 500 GB storage.

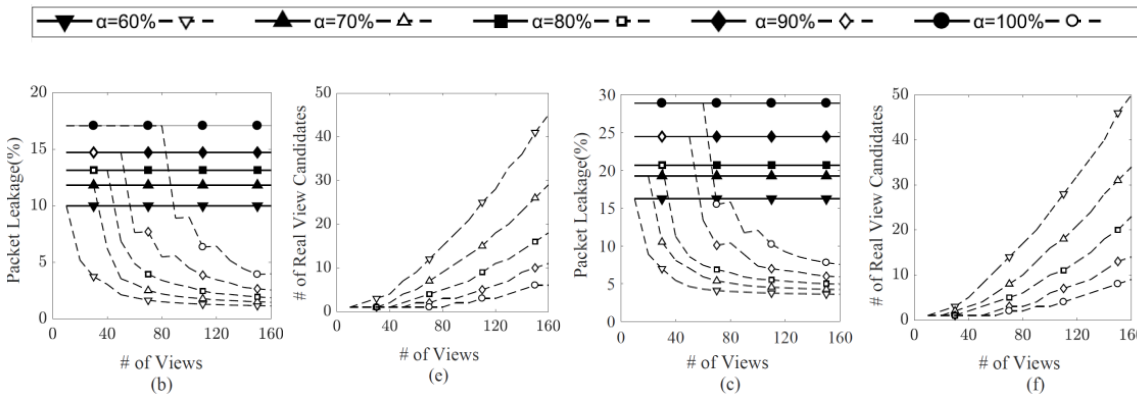


Figure 17: Percentage of the compromised packets in schemes II, III (out of 1M) and number of real view candidates as number of views and the adversarial knowledge vary and for case (1) Figures (b),(e), and (2) Figures (c),(f) where *CP* denotes the CryptoPAn result and *MV* denotes the multi-view results

3.6.2 Information Leakage Analysis

First, the numerical results of the indistinguishability parameter ϵ, and the required number of views to reduce the leakage of CryptoPAn down to 5%, under different levels of adversary knowledge are depicted in Figure 15 (b,c). Those results correspond to three different cases, i.e., when addresses are grouped based on (1) only the first octet (136 groups), (2) the first and the second octets (417 groups), and (3) the first three octets (506 groups). As we can see from the results, ϵ decreases (meaning more privacy) as the number of prefix groups increases, and it increases as the amount of adversarial knowledge increases.

We next validate those numerical results through experiments by measuring the required number of views to reduce the leakage of CryptoPAn down to a certain level. Specifically, we first analyze the behavior of our second and third multi-view schemes (introduced in Sections 3.4.2 and 3.4.3) before comparing the information leakage of the three schemes in Section 3.7. Figure 16 presents different facets of information leakage when our approach is applied in various grouping cases. The results in Figure 16 are for adversaries who has knowledge of no more than 50% of the prefix groups (Figure 17 presents the more extreme cases for the same experiments, i.e., up to 100% knowledge). The analysis of these figures is detailed in the following.

Effect of the number of prefix groups: As discussed earlier, three different IP grouping cases are studied. Figures 16 (a) and (d) show respectively the results of packet leakage and number of real view candidates when $d = 136$. As the numerical results in Figure 9 anticipate, because the fraction $d/D = 0.154$ is relatively low, the indistinguishability of generated views diminishes especially for stronger adversary knowledges. Consequently, the adversary discards more views and the rate of leakage increases, compared with Figures 16 (b), (e) and Figures 16 (c), (f) for which the fraction d/D is 0.47 and 0.57, respectively. In particular, for the worst case of 50% adversary knowledge and when the number of views is less than 50, we can verify that the number of real view candidates for case (1) remains one resulting in packet leakage comparable to that of CryptoPAn.

Effect of the number of views: As it is illustrated in Figure 16, increasing the number of views always improves both the number of real view candidates and the packet leakages. Figure 16 (d-f), show a near linear improvement for real view candidates evaluation, where the slope of this improvement inversely depends on the adversary's knowledge. For the packet leakages, we can observe that the improvement converges to a low packet leakage rate under a large number of views. This is reasonable, as each packet leakage result is an average of leakages in all the real view candidates. However, since each of the fake views leaks a certain amount of information, increasing the number of views beyond a certain value will no longer affect the end result. In other words, the packet leakage converges to the average of leakages in the (fake) real view candidates. Finally, the results show that our proposed scheme can more efficiently improve privacy by (1) increasing the fraction d/D (*number of views/number of distinct addresses*) or (2) increasing the number of views. The first option may affect utility (since inter-group prefix relations will be removed), while the second option is more aligned with our objective of trading off privacy with computation. Specifically, Figure 15 shows that to reduce the leakage down to 10%, generating only 40 views in all the configurations is effective.

Privacy against very strong adversaries: Figure 17 shows the leakage and the real view candidates results for stronger adversaries ($\alpha \in [60, 100]$). Note that Figure 17 only shows results

for case (2) and (3) because our evaluation for case (1) does not show a significant improvement compared with CryptoPAn results due to the fact that the multi-view approach with fraction of $d/D = 0.154$ is not effective against very strong adversaries ($\epsilon > 16$).

3.6.3 Utility Analysis

We now evaluate the utility of the approach using a real network dataset (1M records). For this purpose, we implemented a tool that can parse a network trace, anonymize it w.r.t. an anonymization method from the set of well-known methods including black marker, truncation, random shift [44] (refer to Section 2.1 for details on existing anonymization methods), and our multi-view approach, and finally evaluate the utility of the output using the set of well-known analyses, e.g., IP distribution, packet length distribution, heavy hitter analysis, throughput [125]. To make our discussion more concrete, we consider two disjoint categories of analysis, namely, all types of statistics (1) merely based on the packet characteristics, e.g., timestamp, packet size, etc. or (2) involving IP addresses or subnets that are more important in defining network behavior. Correspondingly, our utility evaluation results are divided into the following two categories.

3.6.3.1 Statistics on fp-QI attributes

Multi-view does not change any of the fp-QI attributes in a network trace, and its methodology focuses on IP addresses, as discussed in Section 5.2. Thus, all kinds of statistics merely depending on fp-QI attributes over each of the generated views are identical to those over the original trace. For instance, Figure 18 shows the results of *empirical cumulative distribution function (CDF)* for the three traces, i.e., (a) the original trace and the real view and (b) one of the fake views. Our results clearly show that all results are identical since multi-view will not have any impact on fingerprinting quasi-identifier attributes.

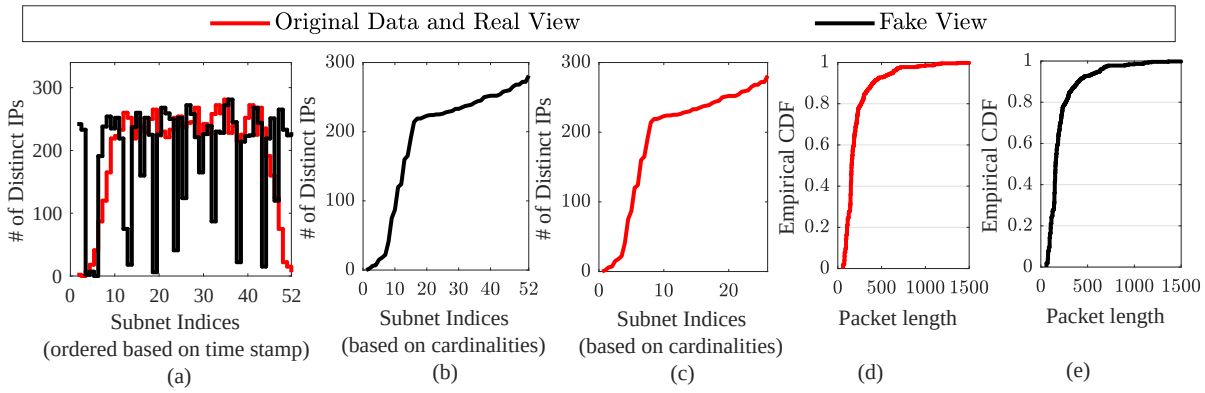


Figure 18: Distribution of distinct IP addresses in different subnets (a-c) (the weighted topology 1) and overall distribution of the packet lengths in (d) the real view and the original trace (e) one of the fake views

3.6.3.2 Statistics related to IP addresses

In the second set of experiments, we study the impact of our multi-view over several types of analyses which are related to IP addresses from the networking literature. We note that while we consider a wide range of analyses, our experiences may not be representative. Moreover, each of our reproductions is only one of many possible ways of reproducing an analysis; different ways of measuring the same quantity may lead to different results. Our evaluation considers four types of weighted IP topology, namely, (1) distribution of distinct IP addresses in different subnets (the weighted topology 1), (2) frequency of the records per subnet (the weighted topology 2), (3) traffic throughput per subnet (the weighted topology 3), and (4) packet throughput per subnet (the weighted topology 4).

Distribution of distinct IP addresses in different subnets: Figure 18 presents a sample IP distribution [88] in the trace, which represents the number of distinct addresses within each subnet (IP group). We compare the distribution of distinct IP addresses inside the original trace, real view and one of the fake views for both *temporal* distribution (if subnets are indexed based on their timestamps), and *cardinality-based* distribution result (if subnets are indexed based on their cardinalities). As shown in Figure 18(a), while the fake view hides this type of IP topology of the

network trace, the real view can preserve the topology since the real view is a prefix preserving mapping of IPs. Moreover, the cardinality-based distribution results, generated from the fake view is identical to those in the original trace and the real view (see Figure 18(b)) which is a result of the imposed indistinguishability in scheme II or III.

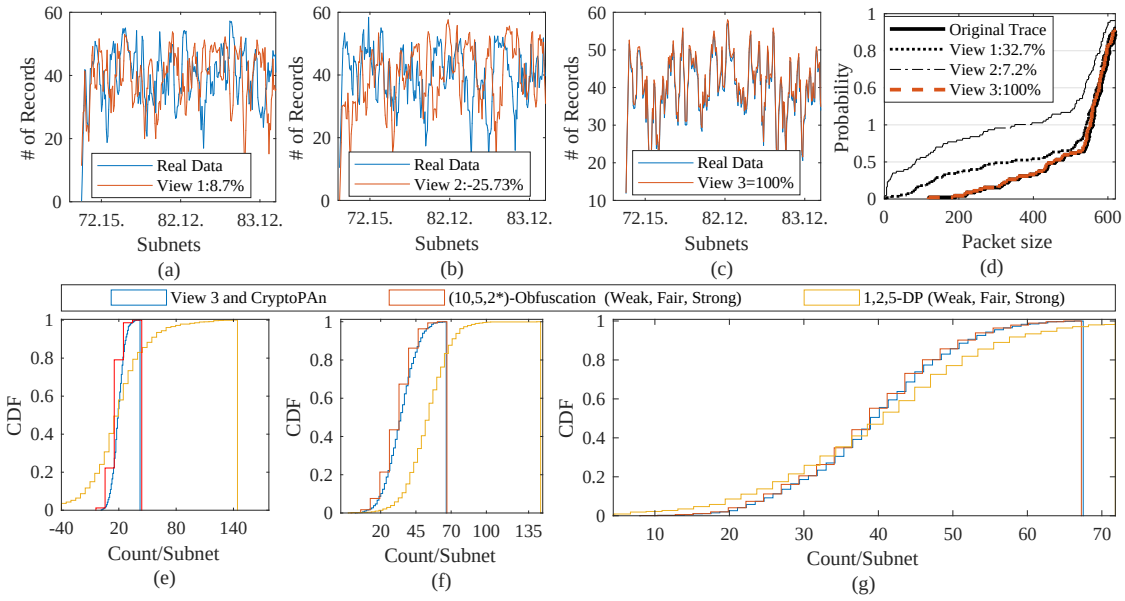


Figure 19: Evaluating frequency of the records per subnet (the weighted topology 2) for different views (a-c), and comparing the CDFs in different views (d), and different privacy metrics (e-g)

Distribution of frequency of the records in different subnets: Figure 19 presents records distribution in a sample collected from the trace, which represents the number of records within each subnet (IP group). This analysis is important in studying various applications, e.g., heavy hitter identification [125], anomaly detection and certain types of attacks like distributed denial of service attack (DDoS) [88]. We compare the results of the distribution for a set of three generated views (a-c) using our multi-view approach where the third view is the real view. We observe that hat, while the real view can completely preserve the statistics, other views incur significant errors. Thus, we conclude that multi-view can also preserve privacy of some types of analysis results. Furthermore, a cumulative distribution function (CDF) of the frequencies is reported in this figure from which we can draw a similar observation. Moreover, Figure 19 (e-g). compare the accuracy

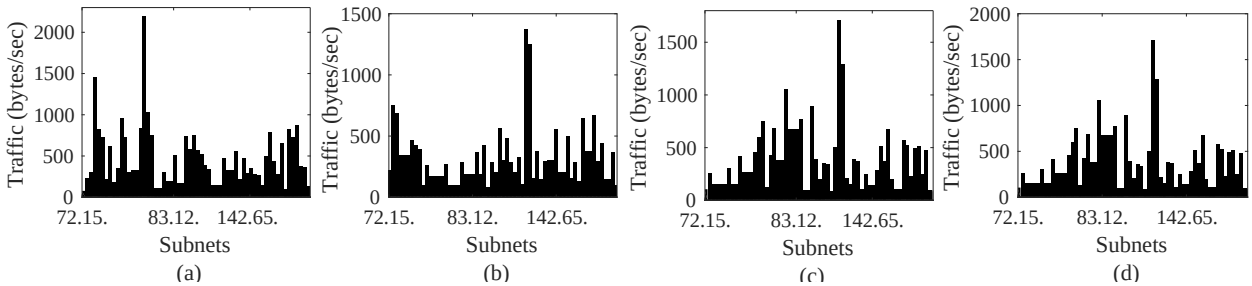


Figure 20: Evaluating the traffic throughput per subnet (the weighted topology 3) for (a),(b) fake views, (c) real view and (d) original trace

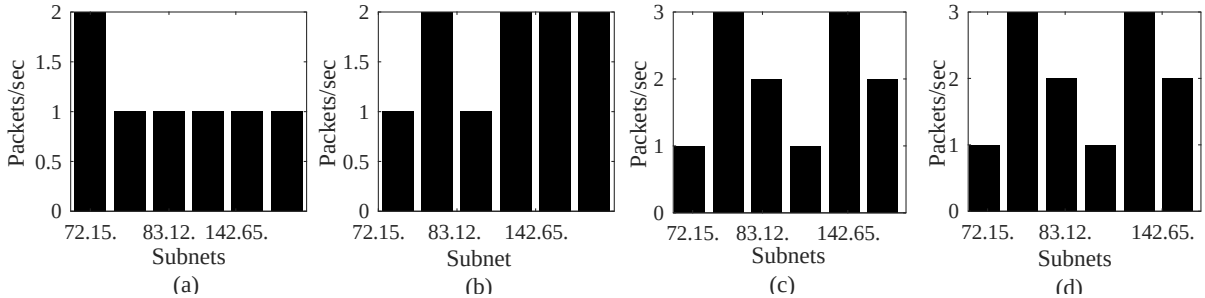


Figure 21: Evaluating the packet throughput per subnet (the weighted topology 4) for (a),(b) fake views, (c) real view and (d) original trace

of IP distribution in (1) the multi-view approach, (2) DP with three categories of privacy guarantees (weak, medium and strong regimes) and (3) (k,j)-obfuscation of Riboni et al. [7]. The results clearly show that for most of analyses our approach outperforms other solutions with relatively weak privacy protections.

Distribution of throughput in different subnets: Figure 20 presents the throughput distribution [88] in a sample of the trace, which represents the rate of the traffic over each subnet (IP group). This analysis is not only important in studying various applications, e.g., heavy hitter identification and computing flow properties such as round trip time (RTT) [88] but also able to reveal confidential information about network activities of the data owners. We compare the distribution results for the three generated views and we observe that, while the real view completely preserves this statistics, other views incur significant errors. Thus, we conclude that multi-view preserves both privacy and utility for this analysis.

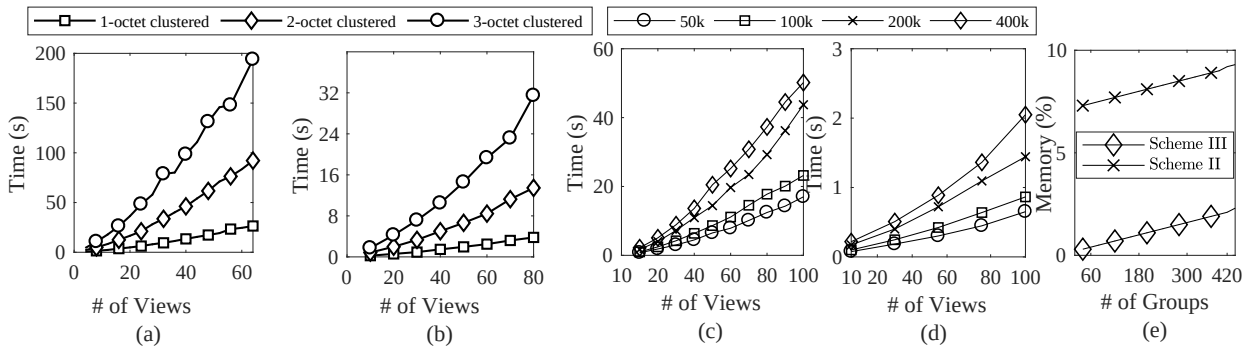


Figure 22: Computation time of schemes II, III for (a,b) different prefix grouping cases, and (c,d) different sizes of dataset, and memory consumption of the two schemes when generating 20 views

Distribution of number of packets per seconds in different subnets: Figure 21 presents the distribution of the number of packets per second in different subnets, which represents the rate of forwarding and receiving packets within each subnet (IP group). All histograms correspond to 2 minutes minute traces collected at the same time of the day. This analysis is important in studying various applications, e.g., certain types of attacks especially, worm fingerprinting [125]. We compare the distribution results for the three generated views. We observe that while the real view completely preserves this statistics, other views incur significant errors. However, the distribution of the packets per seconds in different subnets are designed in a way that the views become indistinguishable.

3.6.4 Performance analysis

In this section, we evaluate the computational cost of our proposed solution in terms of time and memory to investigate its scalability.

Computation time required for generating views: Figure 22 shows the results obtained from running our solution on one million packets while varying the number of generated views. We evaluate the computational overhead incurred by our approach. Figure 22 (a) and (b) show the time required for three different grouping cases under scheme II and scheme III, respectively. We observe that, when the number of views increases, the computational overhead increases nearly

linear. However, each case shows a different slope depending on the number of groups. This is reasonable as our schemes generate key/permutation vectors with a larger number of elements for more groups, which leads to applying CryptoPAn/permutation for a larger number of times. Furthermore, we observe that the time required for scheme III to generate a certain number of views is around 10% of the time for scheme II since applying permutation is much faster than cryptographic operations in CryptoPAn. Finally, linking the results in this figure to the information leakage results shown in Figure 16 demonstrates the trade-off between privacy and computational overhead.

Time and memory overhead for different sizes of data: We evaluate the computation time when varying the size of the dataset. Figure 22 (c), (d) depicts the computation costs of schemes II, III, respectively, when varying the number of views and for different sizes of the dataset. We observe that by increasing the number of views, the computation time increases with a slope depending on the size of the dataset. However, this slope does not increase monotonically. For instance, we observe a bigger jump in the slope of the results of $100k$ to $200k$ than the results of $200k$ to $400k$. This is reasonable as the most important parameter of a dataset in determining the computation time is the number of distinct IP addresses, which does not necessarily increases linearly by size of the dataset. Furthermore, we compare the memory consumption in schemes II and III when generating 20 views. Our results shown in Figure 22 (e) demonstrate that both schemes require a reasonable amount of memory where this the requirement of scheme III is 75% less.

Computation time required for different types of analysis: Finally, we present results on the computation time required for conducting a set of three analyses, when varying the number of views. We find that there is a significant difference between the observations drawn from different types of the analysis. In particular, our results in Table 2 show that the computation time required for conducting an analysis which is merely based on fp-QI, e.g., the overall throughput, is constant for any number of views because the multi-view approach keeps those attributes intact. In contrast,

those analyses that depend on IP addresses require a linearly increasing computation time. However, for those analyses that do not directly depend on IP addresses, e.g., most of the subnet level statistics discussed earlier, the computation time can be saved significantly through techniques like caching. Therefore, the practicality of N times computation will mainly depend on the type of analysis.

Table 2: Computation time per second for different types of analysis

# of Views	Subnet Level	Overall Throughput	IP Level
20	7.62	8.49	71.8
40	11.15	8.49	120.3
60	14.01	8.49	160.34
80	17.42	8.49	205.13

3.6.5 Implementing ORAM in Multi-view

The last step of our solution requires a data owner to privately retrieve an audit report of the real view, which can be based on existing private information retrieval (PIR) techniques. A PIR approach usually aims to conceal the objective of all queries independent of all previous queries [121, 104]. Since the sequence of accesses is not hidden by PIR while each individual access is hidden, the amortized cost is equal to the worst-case cost [121]. Since the server computes over the entire database for each individual query, the results can become impractical. On the other hand, ORAM [79] has verifiably low amortized communication complexity and does not require much computation on the server but rather periodically requires the client to download and reshuffle the data [121]. For our multi-view scheme, we choose ORAM as it is relatively more efficient and secure, and also the client (data owner in our case) has sufficient computational power and storage needed to locally store a small number of blocks (audit reports in our case) in a local stash. In practice, we expect that the analysis reports would have significantly smaller sizes in comparison to the views, and considering the one round communication with ORAM ($O(logN)$-complexity), we believe the solution would have acceptable scalability. Experiments using our dataset and existing ORAM implementation (an implementation [34] of non-recursive Path-ORAM [157] has

been made public) have further confirmed this. We generated various sets of analyses reports using *snort* [32]. Our large-scale experimental dataset only results in *Kilobytes*-level audit reports, which can be practically used with fast ORAM protocols, e.g., Path-ORAM [34]. Specifically, for Path-ORAM, Figure 5 (b) in [34] shows a less than 1MB communication overhead for the worst-case cost of up to 2^{24} number of blocks of size 4KB.

3.7 Discussions

In this section, we discuss various aspects and limitations of our approach.

1. **Exploring background knowledge:** We agree that not all possible attacks on network traces can be addressed by our scheme. As mentioned above, if adversaries possess arbitrary background knowledge, then we would require a DP-based solution, e.g., Mcsherry et al. [36] (which prevents access to raw records and can only support a limited range of analysis). Since this work specifically focuses on network trace anonymization (instead of data privacy in general), we have followed the literature on network trace anonymization to consider the most widely studied threat model (adversarial knowledge of subsets of records or frequency distribution of IP prefixes) as well as utility requirement (releasing a prefix-preserving version of the raw trace) [7, 8, 9, 13, 14, 16, 17, 31]. Such threat model and utility requirement are common since they reflect the most plausible threats in networks (adversaries in a network can either passively observe the traffic in their own subnets, hence the knowledge of subsets of records or frequency distribution, or they can also actively fabricate malicious traffic, hence the injection attack), as well as the common needs for analysis (e.g., analyzing individual records for network anomaly detection or IP traceback). We emphasize that the multi-view solution does not require estimating the exact knowledge (which subsets or prefixes are known) of adversaries, which is certainly impractical, but only the percentage of known data (denoted as alpha-knowledge in this work). Estimating the value of alpha is

similar to estimating the epsilon value of DP, which depends on the desired level of protection (larger alpha means more protection). However, different from most existing solutions (which would provide less utility for more protection), a unique advantage of the multi-view solution is that, as the level of protection increases (with larger alpha), we can still maintain the same level of utility (at the cost of more computation for generating more views).

On the other hand, some attacks are not mentioned in our work but still can be handled by the multi-view solution, e.g., port-based [14], known mapping [16] and machine attribute [17] (those attacks share some characteristics with the injection and fingerprinting attacks by identifying individual machines/IPs in the logs). Moreover, there also exist other attacks which cannot be handled by the existing multi-view solution, but it is feasible to extend the solution to mitigate them. For instance, behavioral profiling on network logs based on distributional adversary knowledge (the overall distribution of certain attribute, e.g., port, and packet size) can be handled by an extended multi-view approach with a redefined partitioning algorithm. Finally, the general idea of multi-view can certainly find applications beyond the domain of network trace protection, and extending the solution with DP is indeed an interesting future direction.

2. **Application to EDB:** We believe the multi-view solution may be applicable to other related areas. For instance, processing on encrypted databases (EDB) has a rich literature including searchable symmetric encryption (SSE) [43], [156], fully-homomorphic encryption (FHE) [75], oblivious RAMs (ORAM) [78], functional encryption [21], and property preserving encryption (PPE) [11], [20]. All these approaches achieve different trade-offs between protection (security), utility (query expressiveness), and computational efficiency [136]. Extending and applying the multi-view approach in those areas could lead to interesting future directions.

3. **Comparing the three schemes:** As we discussed earlier, scheme I achieves a better indistinguishability but less protected partitions in each view. Figure 23 compares the relative

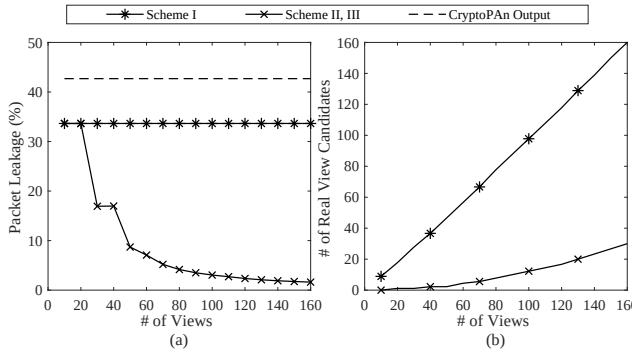


Figure 23: Comparison between the privacy of scheme I vs. schemes II and III with 137 partitions (prefix groups based on the first octet sharing)

effectiveness of the three schemes on a real trace under 40% adversary knowledge. In particular, Figure 23 (a), (b) demonstrate the fact that despite the lower number of real view candidates in schemes II and III compared with scheme I (30 vs 160 out of 160), the end results of the leakage in schemes II and III is much more appealing (3% vs 35%). Therefore, our experimental section has mainly focused on schemes II and III.

4. **Anonymizing more attributes:** In the context of network trace anonymization, IP addresses are generally regarded as the main quasi identifier and have been the central focus of the literature. On the other hand, the multi-view concept (as 2nd layer of protection) can be applied on top of different anonymization techniques (as 1st layer), and not limited to only IP prefix preserving scheme (CryptoPAn) as studied in this work. For instances, since CryptoPAn is a bit-wise operation, it can be potentially extended to other numeric-value attributes, e.g., timestamps, port number and packet sizes, with appropriate indices in order to allow the multi-view solution to work on those attributes.

   As to anonymizing multiple attributes simultaneously, there exists a trade-off between the number of attributes to be anonymized and the overall privacy protection. Specifically, the

more attributes are anonymized, the lower probability for the adversary to successfully identify the records due to more limited information that can be used for reconstruction via injection and fingerprinting. However, he/she with a higher probability can discard fake generated views because the chance of detecting more inconsistencies among correlated attributes can increase.

5. **Choosing the number of views N:** The number of views N is an important parameter of our approach that determines both the privacy and computational overhead. Specifically, as it is implied by Lemma 3.4.2 and demonstrated in our experimental results in Section 3.6, the number of real view candidates is approximately $e^{-\epsilon} \cdot N$. Moreover, the overhead is almost proportional to the number of views because most of the operations are sublinear. However, we stress that in general, generating 100 views can effectively protect the information leakage of CryptoPAn. This reduction squeezes the adversary's post knowledge down to 1-3% of the leakage appears in our baseline, when assuming an unrealistically very strong adversary who has prior knowledge about great number of subnets. Moreover, Figure 22 and Table. 2 present the computation time of view generation and conducting different categories of analysis (both on the analyst's side). These results clearly show the practicality of the Multi-view approach especially when instead of redundantly applying CryptoPAn an efficient rotation of partitions is employed (scheme III). The data owner could choose this value based on the level of trust on the analysts and the amount of computational overhead that can be afforded. An alternative solution is to sacrifice some utility (by giving up some prefix relations among IPs) through increasing the number of prefix groups (D), e.g., grouping them based on the first 3 octets.

6. **Utility:** The main advantage of the multi-view approach is it can preserve the data utility while protecting privacy. In particular, we have shown that the data owner can receive an analysis report based on the real view (Γ_r) which is prefix-preserving over the entire trace.

This is more accurate than the obfuscated (through bucketization and suppression) or perturbed (through adding noise and aggregation) approaches. Specifically, in case of a security breach, the data owner can easily compute $\mathcal{L}_r$ (migration output) to find the mapped IP addresses corresponding to each original address. Then the data owner applies necessary security policies to the IP addresses that are reported violating some policies in Γ_r. A limitation of our work is it only preserve the prefix of IPs, and a potential future direction is to apply our approach to other property-preserving encryption methods such that other properties may be preserved similarly.

7. **Communicational and computational cost:** One of our contributions in this work is to minimize the communication overhead by only outsourcing one (seed) view and some supplementary parameters. This is especially critical for large scale network data like network traces from the major ISPs. On the other hand, one of the key challenges to the multi-view approach is that it requires N times computation for both generating the views and analysis.

   Our experiments in Figure 22 shows that generating 160 views for a trace of $1milion$ packets takes approximately 4 minutes and we describe analytic complexity results in Tables 3 and 4. These tables present overhead analysis, from both the data owner's and the data analyst's side. In particular, table 3 summarizes the overhead for all the action items in the data owner side. Here, $C(n)$ is the computation overhead of CryptoPAn and D is the number of the distinct IP addresses. Finally, table 4 summarizes the overhead for all the action items in the data analyst side where $N \cdot CV(n)$ is the cost of N times verifying the compliances (auditing).

Table 3: Overhead on the data owner side

Blocks in Multi-view	Computation Overhead	Communication Overhead
Initial anonymization	$C(n)$	—
Migration function	$O(nlog^n) + \frac{\sum_{i=1}^{d} C(i)}{d} C(n)$	—
Prefix grouping	—	—
Index generator	$N \cdot O(D)$	$N \cdot O(D)$
Seed trace	$\frac{\sum_{i=1}^{D} V_0(i)}{D} C(n)$	$O(n)$
Report retrieval (ORAM)	—	$O(log^N)\omega(1)$

Table 4: Overhead on the data analyst side

Blocks in Multi-view	Computation Overhead	Communication Overhead
Seed view	—	$O(n)$
N views generation	$\frac{\sum_{i=1}^{N}\sum_{j=1}^{D} V_i(j)}{D} C(n)$	—
Compliance verification (Analysis)	$N \cdot CV(n)$	—

We note that the practicality of N times computation will mainly depends on the type of analysis, and certainly may become impractical for some analyses under large N. How to enable analysts to more efficiently conduct analysis tasks based on multiple views through techniques like caching is an interesting future direction. Another direction is to devise more accurate measures for the data owner to more precisely determine the number of views required to reach a certain level of privacy requirement.

3.8 Summary

In this work, we have proposed a multi-view anonymization approach mitigating the semantic attacks on CryptoPAn while preserving the utility of the trace. This novel approach shifted the trade-off from between privacy and utility to between privacy and computational cost; the later has seen significant decrease with the advance of cloud computing, making our approach a more preferable solution for applications that demand both privacy and utility. We propose three different schemes for our multi-view approach to mitigate different types of attacks. Our experimental results showed that our proposed approach significantly reduced the information leakage compared to CryptoPAn. For example, for the extreme case of adversary pre-knowledge of 100%, the information leakage of CryptoPAn was 100% while under approach it was still less than 10%. Besides addressing various limitations, our future works will adapt the idea to improve existing privacy-preserving solutions in other areas, e.g., we will extend our work to the multi-party problem where several data owners are willing to share their traces to mitigate coordinated network reconnaissance by means of distributed (or inter-domain) audit [31].

Chapter 4

R^2DP: A Universal and Automated Approach to Optimizing the Randomization Mechanisms of Differential Privacy for Utility Metrics with No Known Optimal Distributions

4.1 Introduction

Significant amounts of individual information are being collected and analyzed today through a wide variety of applications across different industries [2]. Differential privacy has been widely recognized as the de facto standard notion [50, 55] in protecting individuals' privacy during such data collection and analysis. On the other hand, since the privacy constraints (e.g., the degree of randomization) imposed by differential privacy may render the released data less useful for analysis, the fundamental trade-off between privacy and utility (i.e., analysis accuracy) has attracted significant attention in various settings [55, 62, 106, 139, 145, 57].

In this context, a key issue is to identify the optimal randomization mechanisms (i.e., distributions and their parameters) [76, 82, 73, 72, 9, 71, 87, 26]). While optimizing the parameters of a given distribution can be easily automated, identifying the optimal distribution for different utility metrics is more challenging, and typically requires manual analysis to examine the search space of all distributions. In fact, recent studies [76, 82, 73, 72, 9, 71, 87, 26] have only identified the optimal randomization mechanisms for a limited number of cases with specific utility metrics and queries. For instance, Ghosh et al. [76, 82] showed that an optimal randomization mechanism (adding a specific class of *geometric noise*) can be used to preserve differential privacy under the class of negative expected loss utility metrics for a single counting query. Subsequently, Geng et al. [73] showed that, under the ℓ_1 and ℓ_2 norms, the widely used standard Laplace mechanism is asymptotically optimal as $\epsilon \to 0$, whereas the Staircase mechanism (which can be viewed as a geometric mixture of uniform probability distributions) performs exponentially better than the Laplace mechanism in case of weaker privacy guarantees (a comprehensive literature review will be given in Section 2.2).

However, this has left the optimal distributions of many other utility metrics as open problems, e.g., usefulness (for machine learning applications [18]), entropy-based measures (for signal processing applications [39, 165], and semi-supervised learning [80]), and graph distance metrics (for social network applications [97]). As shown in the works of Ghosh et al. [76, 82] and Geng et al. [73], different utility metrics will likely lead to different optimal distributions. Moreover, since those existing works mostly rely on manual analysis to examine the search space of all distributions, it would be an expensive process to repeat such efforts for each utility metric. Consequently, many existing works simply employ a well-known distribution (e.g., Laplace noise with constant scale parameter or Gaussian noise with constant variance) without worrying about its optimality. Unfortunately, as our experimental results will show (Section 4.4), choosing a non-optimal distribution (even with its parameters optimized) may lead to rather poor utility.

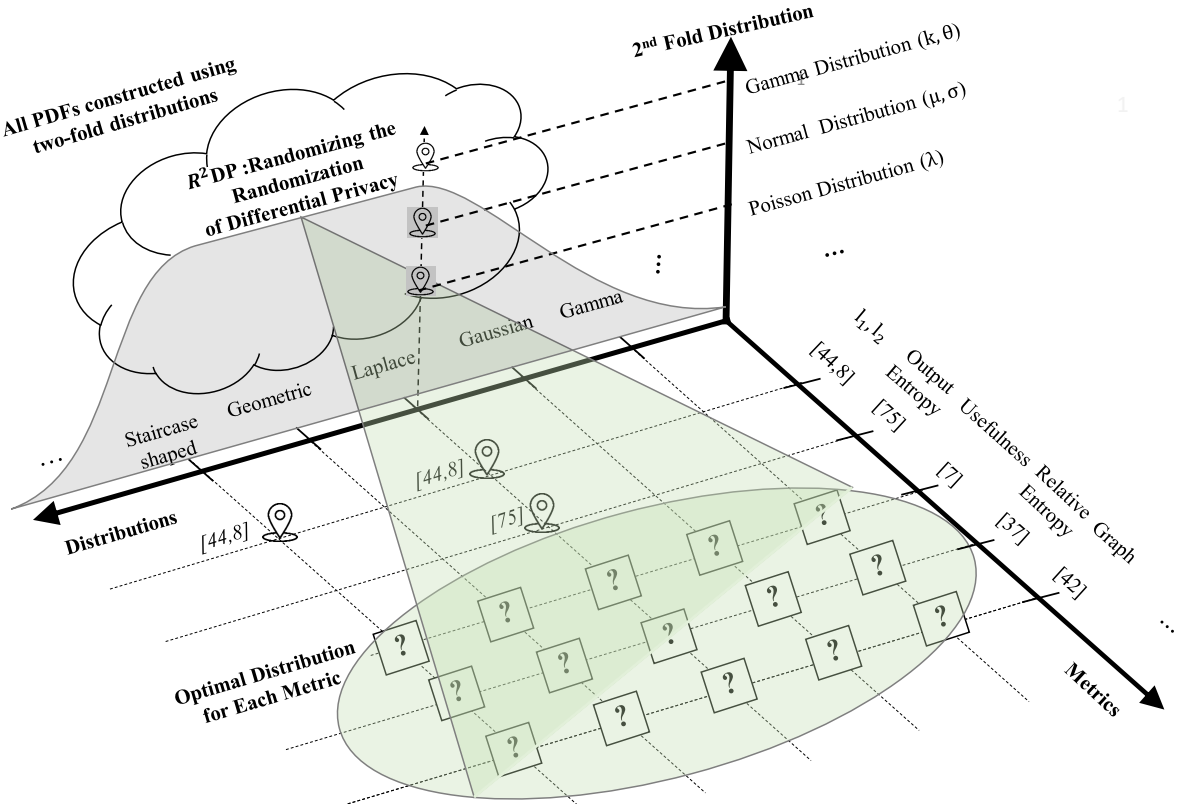


Figure 24: R^{2}DP can automatically optimize different utility metrics which have no known optimal distributions.

4.1.1 R^{2}DP: A Universal Framework

Our key observation is the following. To build a universal framework that can automatically find the optimal distribution in the search space of all distributions, we would need a formulation to link the differential privacy guarantee to the parameters of different distributions (e.g., in Laplace mechanism, ϵ is proportionally related to the inverse of variance). However, it is a known fact that such a formulation varies for each distribution, which explains why existing works have to rely on manual efforts to cover the search space of all distributions, and it also becomes the main obstacle to finding a universal solution that works for all utility metrics employed in different applications.

As depicted in Figure 24, our key idea is that, although it is not possible to directly cover the search space of all distributions in an automated fashion, we can indirectly do so based on the following known fact in probability theory, i.e., a two-fold randomization over the exponential class

of distributions may yield many other distributions to approximately cover the search space [37]. Since this class of distributions are all originated from one of the exponential family distributions, their differential privacy guarantee will become a unique function of the parameters of the second fold distribution. Therefore, these parameters can be used to automatically optimize utility w.r.t. different utility metrics through a universal framework, namely, *r*andomizing the *r*andomization mechanism in *d*ifferential *p*rivacy (R^2DP). Furthermore, the two-fold distribution introduces an added degree of freedom, which allows R^2DP to incorporate the requirements of both data owners and data recipients.

4.1.2 Contributions

Specifically, we make the following contributions:

1. We define the R^2DP framework with several unique benefits. First, it provides the first universal solution that is applicable to different utility metrics, which makes it an appealing solution for applications whose utility metrics have no known optimal distributions (e.g., [18, 39, 165, 97]). Second, unlike most existing works which rely on manual analysis [76, 82], R^2DP can automatically identify a distribution that yields near-optimal utility, and hence is more practical for emerging applications. Third, R^2DP can incorporate the requirements of both data owners and data recipients, which addresses a practical limitation of most existing approaches, i.e., only the privacy budget ϵ is considered in designing the differentially private mechanisms.

2. We formally benchmark R^2DP under the well-studied Laplace mechanism. We tackle several key challenges related to the two-fold distribution in R^2DP. We then show that this mechanism yields a class of log-convex distributions for which the differential privacy guarantee can globally be given in terms of the PDFs' parameters. We also show that it can generate near-optimal results w.r.t. a variety of utility metrics whose optimality is known, e.g., Staircase-shape distribution for large ϵ and Laplace itself for small ϵ [73].

3. We evaluate R^{2}DP using six different utility metrics, both numerically and experimentally on real data, using both statistical queries (e.g., count and average), and data analytics applications (e.g., machine learning and social network). The experimental results demonstrate that R^{2}DP can significantly increase the utility for those utility metrics with no known optimal distributions (compared to the baseline Laplace distribution). We also evaluate the optimality of R^{2}DP using utility metrics whose optimal distributions are known (e.g., Staircase-shape for ℓ_1 and ℓ_2 norms [73]) and our results confirm that R^{2}DP can generate near-optimal results.

4. We discuss the potential of adapting R^{2}DP to improve a variety of other applications related to differential privacy.

The rest of the work is organized as follows. Section 5.2 provides some related background. Section 5.3 defines the R^{2}DP framework. Section 4.3 formally studies the differential privacy guarantee and the utility of R^{2}DP. Section 4.4 presents the experiments. Section 2.3 reviews the related work, and Section 4.6 concludes the work.

4.2 The R^{2}DP framework

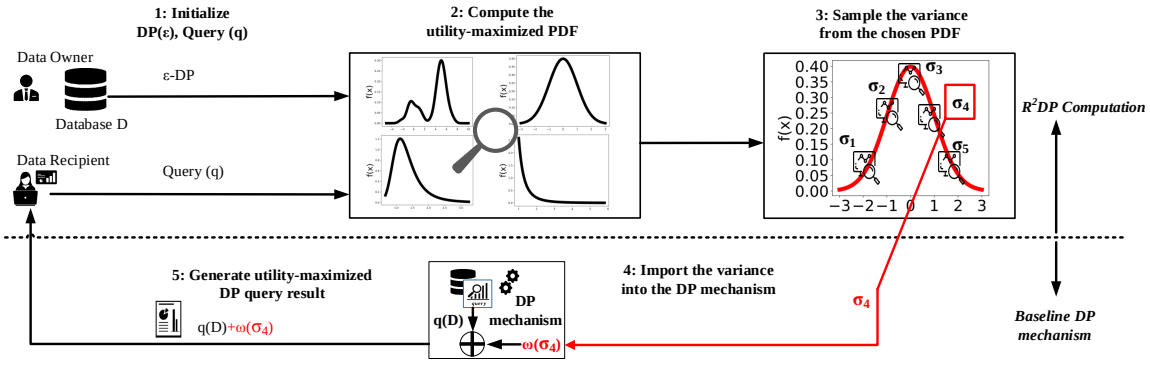


Figure 25: The high level overview of the R^{2}DP framework.

In this section, we define the R^{2}DP framework and its main building block which is the *Utility-maximized PDF* finder.

4.2.1 Notions and Notations

In probability and statistics, a random variable (RV) that is distributed according to some parameterized PDFs, with (some of) the parameters of that PDFs themselves being random variables, is known as a *mixture* distribution [37] when the underlying RV is discrete (or a *compound* distribution when the RV is continuous). Compound (or mixture) distributions have been applied in many contexts in the literature [144] and arise naturally where a statistical population contains two or more sub-populations.

Definition 4.2.1. *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and let X be a RV that is distributed according to some parameterized distribution $f(\theta) \in \mathcal{F}$ with an unknown parameter θ that is again distributed according to some other distribution g. The resulting distribution h is said to be the distribution that results from compounding f with g,*

$$h(X) = \int_{\mathbb{R}} f(X|\theta) g(\theta) \,\mathrm{d}\,\theta \tag{7}$$

Then for any Borel subset B of $\mathbb{R}$,

$$\mathbb{P}(X \in B) = \int_B \int_{\mathbb{R}} f(X|\theta) g(\theta) \,\mathrm{d}\,\theta dX \tag{8}$$

In general, we call any differentially private query answering mechanisms that leverage two-fold probability distribution functions in their randomization, an *R^{2}DP mechanism*.

Definition 4.2.2. *(R^{2}DP Mechanism). Let $\mathcal{M}_q(d, u) = q(d) \bigoplus \omega(u)$ be a mechanism randomizing the answer of a query q using a random oracle $\omega(u)$, where u is the set of parameters (mean, variance, etc.) of the PDF of ω and $\bigoplus$ stands for the corresponding operator. Denote by $\mathcal{F}$ the space of PDFs, we call $\mathcal{M}_q(d, u)$ an R^{2}DP mechanism if at least one of the parameters $u_i \in u$, $(i \leq |u|)$ is/are chosen randomly w.r.t. a specified probability distribution $f_{u_i} \in \mathcal{F}$.*

In particular, the *R^{2}DP Laplace mechanism* will modify the answer to a numerical query by

adding zero-mean noise distributed according to a compound Laplace distribution with the scale parameter b itself distributed according to some distribution f_b.

Example 4.2.1. *Suppose that the scale parameter b in a Laplace mechanism is randomized as follows:*

$$b = \begin{cases} b_1 & \textit{w.p.} \quad p, \\ b_2 & \textit{w.p.} \quad 1-p. \end{cases}$$

Then, the perturbed result $q(D) + Lap(b)$ is an example R^{2}DP Laplace mechanism using a Bernoulli distribution.

Definition 4.2.3. *Let $q : \mathsf{D} \to \mathbb{R}$ be a query and suppose $f_b \in \mathcal{F}$ is a probability density function of the scale parameter b. Then, the mechanism $\mathcal{M}_q : \mathsf{D} \times \Omega \to \mathbb{R}$, defined by $\mathcal{M}_q(d, b) = q(d) + Lap(b)$ is an R^{2}DP Laplace mechanism that utilizes PDF f_b.*

4.2.2 The Framework

As shown in Figure 25, R^{2}DP framework include the following steps.

R^{2}DP Computation:

- **Step 1**: The data owner specifies the differential privacy budget ϵ and the data recipient specifies his/her query of interest together with its required utility metric.
- **Step 2**: Given the input triplets (ϵ, query, metric), the *utility-maximized PDF computing module* computes the provably optimal probability density function and its parameters for the variance of the additive noise. For example, in Figure 25, the PDF computing module returns a lower tail truncated Gaussian distribution for the specified inputs.
- **Step 3**: The *variance sampler* module randomly samples (w.r.t. the PDF found in Step 2) one standard deviation σ_i of the noise to be eventually added.

Baseline DP Randomization:

- **Step 4**: Next, the computed standard deviation σ_i is used to generate a noise $\omega(\sigma_4)$ for the baseline DP mechanism, which is a DP mechanism of exponential order, e.g ., Laplace, Gaussian and exponential mechanisms.

- **Step 5**: The computed noise $\omega(\sigma_i)$ is added to the query result $q(D)$ to provide a utility-maximized DP result to the data recipient.

The most important module of the R^{2}DP framework is the *utility-maximized PDF computing module* (Step 2) which will be described in more details in the following. Furthermore, to make our discussions more concrete, we instantiate the R^{2}DP framework based on the well studied Laplace mechanism, namely, *the R^{2}DP Laplace mechanism*, where other baseline DP mechanisms will be discussed in Appendix 4.5.6 due to space limitation (from now on, we will simply refer to the R^{2}DP Laplace mechanism as R^{2}DP). Particularly, we show that, with a two-fold Laplace distribution, an infinite-size class of log-convex distributions can be identified. This class of distributions pertains a differential privacy guarantee which can globally be given in terms of the PDFs' parameters, and hence is automatically optimizable under the differential privacy constraint.

4.2.3 Computing Utility-Maximized PDF

In Figure 25, to compute the utility-maximized PDF (Step 2), a key challenge is to establish the search space of automatically optimizable PDFs, from which the utility-maximized PDF is computed. Ideally, the search space of an R^{2}DP mechanism can be defined as the collection of all two-fold distributions, e.g., with Laplace and exponential as the first and second fold distributions, respectively. However, the key challenge here is that a mixture of distributions is itself a distribution which does not necessarily provide a global differential privacy guarantee in terms of the resulting PDFs' parameters (automatically optimizable under the differential privacy constraint). To address this issue, the *Moment Generating Function (MGF)* [66] of the second fold distribution could be utilized, e.g., given the first fold as Laplace distribution. Specifically, MGF of a random variable is an alternative specification of its probability distribution, and hence provides the basis

of an alternative route to analytical results compared with directly using probability density functions or cumulative distribution functions [66]. In particular, the MGF of a random variable is a log-convex function of its probability distribution which can provide a global differential privacy guarantee [66] (see Theorem 4.3.1).

Definition 4.2.4. *(Moment Generating Function [66]). The moment-generating function of a random variable x is $M_X(t) := \mathbb{E}\left[e^{tX}\right], t \in \mathbb{R}$ wherever this expectation exists. The moment-generating function is the expectation of the random variable e^{tX}.*

Theorem 4.2.1. *We can write the CDF of the output of an R^{2}DP mechanism in terms of the* Moment Generating Function (MGF) *[66] of the probability distribution $f_{\frac{1}{b}}$, where b is the randomized scale parameter (see Appendix 4.5.1 and 4.5.3 for the details and the proof).*

Thus, for a PDF with non-negative support (since scale parameter is always non-negative), the R^{2}DP mechanism outputs another PDF using the MGF (where CDF is the moment and PDF is its derivative, as shown in Equation 15 in Appendix 4.5.3) . Moreover, since MGF is a bijective function [65], the R^{2}DP mechanism can in fact generate a search space as large as the space of all PDFs with non-negative support and an existing MGF. However, the next challenge is that not all random variables have moment generating functions (MGFs), e.g., Cauchy distribution [28]. Fortunately, MGFs possess an appealing composability property between independent probability distributions [37], which can be used to provide a search space of all linear combinations of a set of popular distributions with known MGFs (infinite number of RVs).

Theorem 4.2.2 (MGF of Linear Combination of RVs)**.** *If $x_1, \cdots, x_n$ are n independent RVs with MGFs $M_{x_i}(t) = \mathbb{E}(e^{tx_i})$ for $i = 1, \cdots, n$, then the MGF of the linear combination $Y = \sum_{i=1}^{n} a_i x_i$ is* $\prod_{i=1}^{n} M_{x_i}(a_i t)$.

Consequently, we define the search space of the R^{2}DP mechanism as all possible linear combinations of a set of independent RVs with existing MGF (Section 4.3.2.2 will provide more details on how to choose the set of independent RVs). Although this search space is only a subset of all

two-fold distributions, we will show through both numerical results (in Section 4.5.5) and experiments with real data (Section 4.4) that this search space is indeed sufficient to generate near-optimal utility w.r.t. all utility metrics (universality).

4.3 Privacy and Utility

In this section, we analyze the privacy and utility of the R^{2}DP, and then discuss extensions for improving and implementing R^{2}DP.

4.3.1 Privacy Analysis

We now show the R^{2}DP mechanism provides differential privacy guarantee. By Theorem 4.2.1, the DP bound of the R^{2}DP is

$$e^{\epsilon} = \max_{\forall S \in \mathsf{R}} \left\{ \frac{-M_{\frac{1}{b}}(-|x-q(d)|)|_{S_{\geq q(d)}} + M_{\frac{1}{b}}(-|x-q(d)|)|_{S_{<q(d)}}}{-M_{\frac{1}{b}}(-|x-q(d')|)|_{S_{\geq q(d')}} + M_{\frac{1}{b}}(-|x-q(d')|)|_{S_{<q(d')}}} \right\}$$

Hence, the value of e^{ϵ} only depends on the distribution of reciprocal of the scale parameter b, i.e., $f_{\frac{1}{b}}$. Moreover, an MGF is positive and log-convex [66] where the latter property is desirable in defining various natural logarithm upper bounds, e.g., DP bound. In the following theorem, our MGF-based formula for the probability $\mathbb{P}(\{q(d) + Lap(b)\} \in S)$ can be easily applied to calculate the differential privacy guarantee (see Appendix 4.5.3 for the proof).

Theorem 4.3.1. *The R^{2}DP mechanism $\mathcal{M}_q(d, b)$ is*

$$\ln \left[\frac{\mathbb{E}(\frac{1}{b})}{\frac{\mathrm{d}M_{\frac{1}{b}}(t)}{\mathrm{d}t}|_{t=-\Delta q}} \right] \text{-}\textit{differentially private.} \tag{9}$$

Moreover, Theorem 4.2.2 can be directly applied to calculate the differential privacy guarantee of any RV from the search space defined in Section 4.2.3 (i.e., all linear combinations of a set of

independent RVs with known MGFs).

Corollary 4.3.1 (Differential Privacy of Combined PDFs)**.** *If $x_1, \cdots, x_n$ are n independent random variables with respective MGFs $M_{x_i}(t) = \mathbb{E}(e^{tx_i})$ for $i = 1, \cdots, n$, then the R^{2}DP mechanism $\mathcal{M}_q(d, b)$ where $\frac{1}{b}$ is defined as the linear combination $\frac{1}{b} = \sum_{i=1}^{n} a_i x_i$ is ϵ-differentially private, where*

$$\epsilon = \ln \left[\frac{\sum_{j=1}^{n} a_j \cdot E_{x_j}(\frac{1}{b})}{\sum_{j=1}^{n} a_j \cdot M'_{x_j}(-a_j \cdot \Delta q) \cdot \prod_{\substack{i=1 \\ i \neq j}}^{n} M_{x_i}(-a_i \cdot \Delta q)} \right] \tag{10}$$

Therefore, we have established a search space of probability distributions with a universal formulation for their differential privacy guarantees, which is the key enabler for the universality of R^{2}DP. Next, we characterize the utility of R^{2}DP mechanisms.

4.3.2 Utility Analysis

We now characterize the utility of the R^{2}DP mechanism. To make concrete discussions, we focus on the usefulness metric (see Section 1.5), and a similar logic can also be applied to other metrics.

4.3.2.1 Characterizing the Utility

Denote by $U(\epsilon, \Delta q, \gamma)$ the usefulness of an R^{2}DP mechanism for all $\epsilon > 0$, sensitivity Δq and error bound γ. The optimal usefulness is then given as the answer of the following optimization problem over the search space of PDFs.

$$\max_{f_{\frac{1}{b}} \in F} \{U(\epsilon, \Delta q, \gamma)\} = \max_{f_{\frac{1}{b}} \in F} \Bigg\{ \tfrac{1}{2} \cdot \Big[- M_{\frac{1}{b}}(-|x - q(d)|)|_{q(d)}^{q(d)+\gamma} + M_{\frac{1}{b}}(-|x - q(d)|)|_{q(d)-\gamma}^{q(d)} \Big] \Bigg\},$$

$$\text{subject to} \quad \epsilon = \ln \left[\frac{\mathbb{E}(\frac{1}{b})}{\frac{\mathrm{d}M_{\frac{1}{b}}(t)}{\mathrm{d}t}|_{t=-\Delta q}} \right]$$

where the utility function is the probability of generating ϵ-DP query results within a distance of γ-error (using Theorem 4.2.1). Note that ϵ and Δq do not directly impact the usefulness but they do so indirectly through the differential privacy constraint. Furthermore, as shown in Theorem 4.3.1, the differential privacy guarantee ϵ over the established search space is a unique function of the parameters of the second-fold distribution.

Corollary 4.3.2. *Denote by u, the set of parameters for a probability distribution $f_{\frac{1}{b}}$, and by $M_{f(u)}$ its MGF. Then, the optimal usefulness of an R^{2}DP mechanism utilizing $f_{\frac{1}{b}}$, at each triplet $(\epsilon, \Delta q, \gamma)$ is*

$$U_f(\epsilon, \Delta q, \gamma) = \max_{u \in \mathbb{R}^{|u|}} \Bigg\{ \tfrac{1}{2} \cdot \Big[- M_{f(u)}(-|x - q(d)|)|_{q(d)}^{q(d)+\gamma} + M_{f(u)}(-|x - q(d)|)|_{q(d)-\gamma}^{q(d)} \Big] \Bigg\},$$

$$\textit{subject to} \quad \epsilon = \ln \left[\frac{\mathbb{E}(\frac{1}{b})}{\frac{\mathrm{d}M_{\frac{1}{b}}(t)}{\mathrm{d}t}|_{t=-\Delta q}} \right]$$

Since MGFs are positive and log-convex, with $M(0) = 1$, we have $U_f(\epsilon, \Delta q, \gamma) = 1 - \min_{u \in \mathbb{R}^{|u|}} M_{f(u)}(-\gamma)$. Thus, for usefulness metric, the optimal distribution for ϵ is the one with the minimum MGF evaluated at γ. In particular, for a set of privacy/utility parameters, we can find the optimal PDF using the *Lagrange multiplier* [12]. i.e.,

$$\mathcal{L}(u,\lambda) = M_{f(u)}(-\gamma) + \lambda \cdot (\ln \left[\frac{\mathbb{E}(\frac{1}{b})}{\frac{\mathrm{d}M_{\frac{1}{b}}(t)}{\mathrm{d}t}|_{t=-\Delta q}} \right] - \epsilon) \tag{11}$$

Moreover, Theorem 4.2.2 can be directly applied to design a utility-maximizing R^2DP mechanism with a sufficiently large search space (with an infinite number of different random variables).

Corollary 4.3.3 (Optimal Utility for Combined RVs)**.** *If $x_1, x_2, \cdots, x_n$ are n independent random variables with respective MGFs $M_{x_i}(t) = \mathbb{E}(e^{tx_i})$ for $i = 1, 2, \cdots, n$, then for the linear combination $Y = \sum_{i=1}^{n} a_i x_i$, the optimal usefulness (similar relation holds for other metrics) under ϵ-differential privacy constraint is given as*

$$U_Y(\epsilon, \Delta q, \gamma) = 1 - \min_{\mathcal{A},\mathcal{U}} \left\{ \prod_{i=1}^{n} M_{x_i}(-a_i\gamma) \right\} \tag{12}$$

subject to

$$\epsilon = \ln \left[\frac{\sum_{j=1}^{n} a_j \cdot E_{x_j}(\frac{1}{b})}{\sum_{j=1}^{n} a_j \cdot M'_{x_j}(a_j \cdot -\Delta q) \cdot \prod_{\substack{i=1 \\ i\neq j}}^{n} M_{x_i}(-a_i \cdot \Delta q)} \right]$$

where $\mathcal{A} = \{a_1, a_2, \cdots, a_n\}$ is the set of the coefficients and $\mathcal{U} = \{u_1, u_2, \cdots, u_n\}$ is the set of parameters of the probability distributions of RVs x_i, $\forall i \leq n$.

Similar to the case of a single RV, we can compute the optimal solution for this optimization problem using the Lagrange multiplier function in Equation 11.

4.3.2.2 Finding Utility-Maximizing Distributions

Since not all second-fold probability distributions can boost the utility of the baseline Laplace mechanism, leveraging all RVs into our search space would only result in redundant computation by the utility-maximized PDF computing module. Accordingly, in this section, we first derive a

necessary condition on the differential privacy guarantee of R^{2}DP to boost the utility of the baseline Laplace mechanism (refer to Appendix 4.5.3 for the proof). Using this necessary condition, we can easily filter out those probability distributions that cannot deliver any utility improvement.

Theorem 4.3.2. *The utility of R^{2}DP with $\epsilon \geq \ln\left[\mathbb{E}_{\frac{1}{b}}\left(e^{\epsilon(b)}\right)\right]$ is always upper bounded by the utility of the ϵ-differentially private baseline Laplace mechanism. Equivalently, for an R^{2}DP mechanism to boost the utility, the following relation is necessarily true.*

$$e^{\epsilon} = \frac{\mathbb{E}(\frac{1}{b})}{M'_{\frac{1}{b}}(-\Delta q)} < M_{\frac{1}{b}}(\Delta q) \tag{13}$$

We note that $\epsilon = \ln\left[\mathbb{E}_{\frac{1}{b}}\left(e^{\epsilon(b)}\right)\right]$ provides a tight upper bound since it gives the overall e^{ϵ} of an R^{2}DP mechanism as the average of differential privacy leakage. Next, we examine a set of well-known PDFs as second-fold distribution to identify the distribution that offers a significantly improved utility compared with the bound given in Theorem 4.3.2. Promisingly, our analytic evaluations for *three* of these distributions, i.e., Gamma, uniform and truncated Gaussian distributions demonstrate such a payoff (Appendix 4.5.2 theoretically analyzes several case study PDFs). We note that those chosen distributions are general enough to cover many of other probability distributions (e.g., Exponential, Erlang, and Chi-squared distributions are special cases of Gamma distribution).

4.3.2.3 Deriving Error Bounds

The error bounds of the R^{2}DP mechanism under some well-known utility metrics are shown in Table 9. The key idea in deriving these results is to calculate the mean of each utility metric over the PDF of RV $1/b$ (which is the linear combination of RVs in multiple PDFs). Specifically, given the error bound $e_L(b)$ for deterministic variance (i.e., Laplace mechanism), the total error bound of an R^{2}DP mechanism will be the mean $\int_0^{\infty} e_L(b) f_b(b) db$. The results shown in Table 9 can be easily

applied to optimize those metrics in corresponding applications (e.g., ℓ_1 for private record matching [91], ℓ_2 for location privacy [22], usefulness for machine learning [18], Mallows for social network analysis [89], and relative entropy (with a degree α) for semi-supervised learning [80]).

Table 5: Error bound of R^{2}DP under different metrics

Metric	Dependency to Prior	R^{2}DP Error Bound
ℓ_1	independent	$\int_0^\infty M_{\frac{1}{b}}(-x)dx$
ℓ_2	independent	$\sqrt{2\int_0^\infty\int M_{\frac{1}{b}}(-u)dudx}$
Usefulness	independent	$1 - M_{\frac{1}{b}}(-\gamma)$
Mallows (p)	dependent	$\left([\sum_{i=1}^n \|N_i \sim [-M'_{\frac{1}{b}}(-x)/2]\|^p]/n\right)^{1/p}$
Relative Entropy (α)	dependent	$\frac{\log\sum_{x\in\mathcal{X}} p(x)^\alpha q(x)^{1-\alpha}}{\alpha-1}$ s.t$(q(x)-p(x)) \sim -M'_{\frac{1}{b}}(-x)/2$

In this context, the ℓ_1, ℓ_2 and usefulness metrics (as defined in Section 1.5) are independent to the prior (i.e., not depending on the distribution of the true results). The metrics will be evaluated based on the deviation between the true and noisy results (which does not change regardless of the prior). On the contrary, some other metrics (e.g., Mallows and relative entropy) depend on the prior distribution of the true results [89, 80]. In such cases, the metrics will be evaluated based on the deviation between the true and noisy results w.r.t. the prior in specific experimental settings (we will discuss those specific priors used in the experiments in Section 4.4).

Table 6: R^{2}DP compared to Laplace w.r.t. error bounds for learning algorithms

	Linear SVM [94]	Bayesian Inference (statistician) [182]	Robust Linear Regression [54]	Naive Bayes [161]
Laplace	$O(\frac{\log(1/\beta)}{\alpha^2} + \frac{1}{\epsilon\alpha} + \frac{\log(1/\beta)}{\alpha\epsilon})$	$O(mn\log(n))[1 - exp(-\frac{n\epsilon}{2\|\mathcal{I}\|}]$	$O(n^{-\epsilon log n})$	$O(\frac{1}{n\epsilon})$
R^{2}DP	$O(\frac{\log(1/\beta)}{\alpha^2} + \mathbb{E}_{\frac{1}{b}}(\frac{b}{\alpha} + \frac{b\log(1/\beta)}{\alpha}))$	$O(mn\log(n))[1 - M_{\frac{1}{b}}(-\frac{n}{2\|\mathcal{I}\|})]$	$O(\mathbb{E}_{\frac{1}{b}}(n^{-\frac{log n}{b}}))$	$O(\mathbb{E}_{\frac{1}{b}}(\frac{b}{n}))$

In addition to the error bounds given in Table 9, an analyst can derive error bounds for more advanced queries, e.g., those pertaining to learning algorithms [94, 182, 54, 161]. Given the error bound of Laplace mechanism in an application (e.g., Linear SVM [94]), the error bound of the R^{2}DP framework for this application can be derived by taking average of the Laplace's result over the PDF of $\frac{1}{b_r}$. In particular, Table 6 demonstrates the error bounds of R^{2}DP for some learning algorithms (as shown in Section 4.4, those learning algorithms can benefit from integrating R^{2}DP instead of Laplace).

To derive the error bounds shown in Table 9 and Table 6, the noise parameter(s) and the PDFs used in R^{2}DP can be released to a downstream analyst. This will not cause any privacy leakage

because, similar to other differential privacy mechanisms, the privacy protection of R2DP comes from the (first-fold) randomization (whose generated random noises are never disclosed), which will not be affected even if all the noise parameter(s) and the PDFs are disclosed (see Section 4.3.1 and Appendix 4.5.3 for the formal privacy analysis and proof). We note that, although R^{2}DP replaces the fixed variance of a standard differential privacy mechanism with a random variance, this second-fold randomization is not meant to keep the generated parameters (e.g., the variance) secret, but designed to cover a larger search space (as detailed in Section 4.2.3).

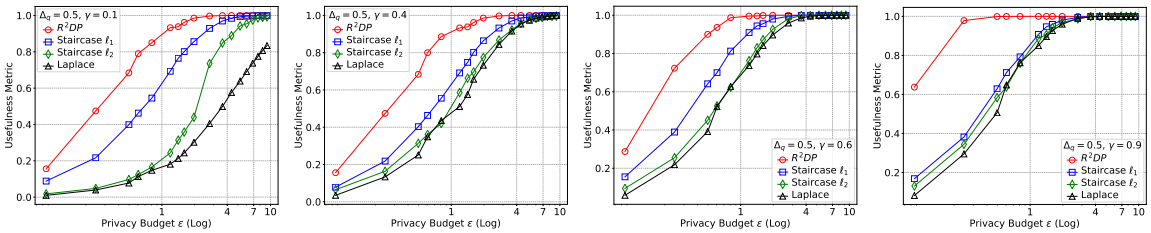
4.3.3 R^{2}DP Algorithm

Algorithm 2 details an instance of the R^{2}DP framework using linear combination of three different PDFs. In particular, the algorithm with ϵ-DP finds the best second-fold distribution using the Lagrange multiplier function (see Appendix 4.5.4) that optimizes the utility metric. Then, it randomly generates the noise using the two-fold distribution (e.g., first-fold Laplace) and injects it into the query.

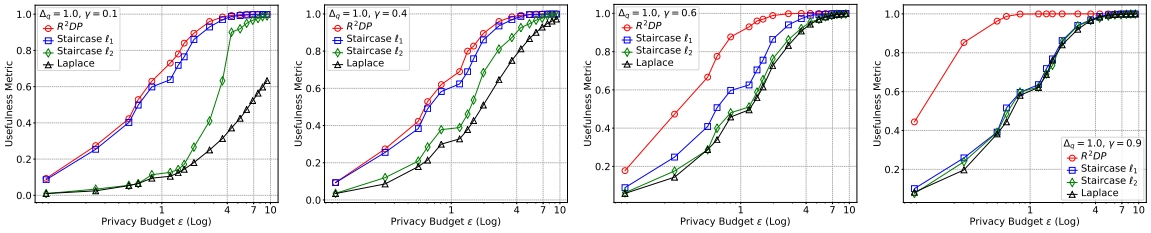
Input : Dataset D, Privacy budget ϵ, Query $q(\cdot)$, Metric and its parameters (from data recipient)
Output: Query result $q(D) + Lap(b_r)$, DP guarantee ϵ, Second-fold PDF's parameters
$\Delta q \leftarrow$ Sensitivity $(q(\cdot))$
Find optimal parameters from Lagrange Multiplier $\mathcal{L}(\epsilon, \Delta q, \text{metric}) =$
$a_1^{opt}, a_2^{opt}, a_3^{opt}, k^{opt}, \theta^{opt}, a_u^{opt}, b_u^{opt}, \mu^{opt}, \sigma^{opt}, a_{\mathcal{N}^T}^{opt}, b_{\mathcal{N}^T}^{opt}$
$X_1 \sim \Gamma(k^{opt}, \theta^{opt})$
$X_2 \sim U(a_u^{opt}, b_u^{opt})$
$X_3 \sim \mathcal{N}^T(\mu^{opt}, \sigma^{opt}, a_{\mathcal{N}^T}^{opt}, b_{\mathcal{N}^T}^{opt})$
$\frac{1}{b_r} = a_1^{opt} \cdot X_1 + a_2^{opt} \cdot X_2 + a_3^{opt} \cdot X_3$
return $q(D) + Lap(b_r)$, ϵ, $\mathcal{L}(\epsilon, \Delta q, \text{metric})$

Algorithm 2: The Ensemble R^{2}DP Algorithm

Some advanced applications (e.g., workload queries) that integrate R^{2}DP to improve their utility are discussed in Appendix 4.5.9.



(a) $\Delta q = 0.5, \gamma = 0.1$ (b) $\Delta q = 0.5, \gamma = 0.4$ (c) $\Delta q = 0.5, \gamma = 0.6$ (d) $\Delta q = 0.5, \gamma = 0.9$



(e) $\Delta q = 1, \gamma = 0.1$ (f) $\Delta q = 1, \gamma = 0.4$ (g) $\Delta q = 1, \gamma = 0.6$ (h) $\Delta q = 1, \gamma = 0.9$

Figure 26: Usefulness metric: R^{2}DP (with five PDFs, i.e., Gamma, Uniform, Truncated Gaussian, Noncentral Chi-squared and Rayleigh distributions) strictly outperforms Laplace and Staircase mechanisms for statistical queries, where the ratio of improvement depends on the values of Δq, γ and ϵ.

4.4 Experimental Evaluations

In this section, we experimentally evaluate the performance of R^{2}DP using six different utility metrics, i.e., ℓ_1, ℓ_2, entropy, usefulness, Mallows and Rényi divergence. Furthermore, we investigate the tightness of R^{2}DP under Rényi differential privacy (RDP in short) [129] which provides a universal formulation of the privacy losses of various DP mechanisms, as shown in Appendix 4.5.8.2 (facilitating the comparison between different mechanisms). Our objective is to verify the following two properties about the performance of the R^{2}DP framework w.r.t. all seven utility and privacy metrics: (1) R^{2}DP produces near-optimal results and (2) R^{2}DP performs strictly better than well-known baseline mechanisms, e.g, Laplace and Staircase mechanisms, in settings where an optimal PDF is not known, e.g., usefulness utility metric or Rényi differential privacy .

4.4.1 Experimental Setting

We perform all the experiments and comparisons on the Privacy Integrated Queries (PINQ) platform [123]. Besides basic statistical queries, two applications in the current suite (*machine learning* and *social network analysis*) are employed to evaluate the accuracy of R^{2}DP and compare it to Laplace and Staircase mechanisms. I would like to thank Shangyu Xie for his help in conducting the following results.

4.4.1.1 Statistical Queries

In the first set of our experiments, we examine the benefits of R^{2}DP using basic statistical functions, i.e., count and average. The dataset comes from a sensor network experiment carried out in the Mitsubishi Electric Research Laboratories (MERL) and described in [168]. MERL has collected motion sensor data from a network of over 200 sensors for a year and the dataset contains over 30 million raw motion records. To illustrate the query performance with different sensitivities, we create the queries based on a subset of the data including aggregated events that are recorded by closely located sensors over 5-minute intervals. We formed in this way 10 input signals corresponding to 10 spatial zones (each zone is covered by a group of sensors). Since each individual can activate several sensors and travel through different zones, we define moving average functions with arbitrary sensitivity values, e.g., $\Delta q \in [0.1, 5]$. For instance, we could be interested in the summation of the moving averages over the past 30 min for zones 1 to 4. We apply R^{2}DP w.r.t. usefulness, ℓ_1, ℓ_2, entropy, and Rényi metrics, respectively.

4.4.1.2 Social Network

Social network degree distribution is performed on a Facebook dataset [107]. They consist of "circles" and "friends lists" from Facebook by representing different individuals as nodes (47,538 nodes) and friend connections as edges (222,887 edges). Recall that the Mallows metric is frequently used for social network (graph-based) applications [97]. We thus apply R^{2}DP w.r.t. the

Mallows metric in this group of experiments.

4.4.1.3 Machine Learning

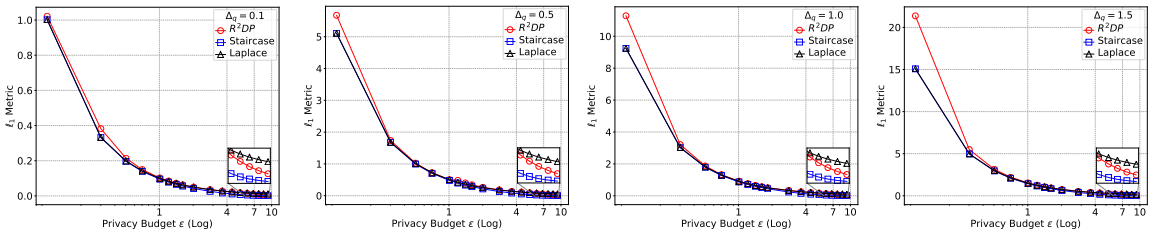
Naive Bayes classification is performed on two datasets: Adult dataset (in the UCI ML Repository) [101] and KDDCup99 dataset [158]. First, the Adult dataset includes the demographic information of 48,842 different adults in the US (14 features). It can be utilized to train a Naive Bayes classifier to predict if any adult's annual salary is greater than 50k or not. Second, the KDD competition dataset was utilized to build a network intrusion detector (given 24 training attack types) by classifying "bad" connections and "good" connections. Recall that the usefulness metric is commonly used for machine learning [18]. We thus apply R^{2}DP w.r.t. the usefulness in this group of experiments.

4.4.2 Basic Statistical Queries

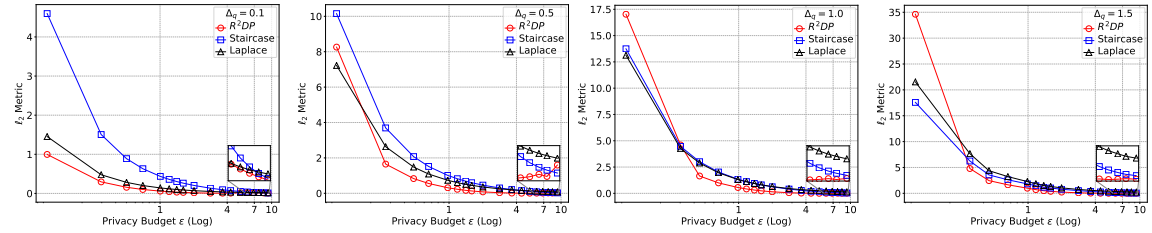
We validate the effectiveness of R^{2}DP using two basic statistical queries: count (sensitivity=1) and moving average with different window sizes, e.g., sensitivity $\in [0.1, 2]$ to comprehensively study the performance of R^{2}DP by benchmarking with Laplace and Staircase mechanisms. We have the following observations.

4.4.2.1 Usefulness Metric

We compare R^{2}DP with the baseline Laplace and two classes of Staircase mechanisms proposed in [82] w.r.t. ℓ_1 and ℓ_2 metrics, by varying the privacy budget ϵ, four error bounds $\gamma \in \{0.1, 0.4, 0.6, 0.9\}$ and two different sensitivities (Section 4.5.5 additionally shows numerical results to provide a more comprehensive evaluation for the usefulness metric). As shown in Figure 26, R^{2}DP generates strictly better results w.r.t. the usefulness metric, and the ratio of improvement depends on values of ϵ, Δq and γ. In particular, we observe that the improvement is relatively larger for a larger error bound and smaller sensitivity (Figure 26 (a,b,e,f) vs. (c,d,g,h)). One important factor determining

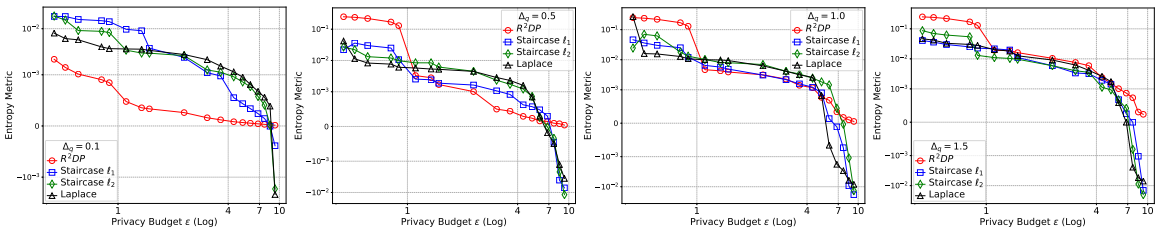


(a) $\ell_1, \Delta q = 0.1$ (b) $\ell_1, \Delta q = 0.5$ (c) $\ell_1, \Delta q = 1$ (d) $\ell_1, \Delta q = 1.5$



(e) $\ell_2, \Delta q = 0.1$ (f) $\ell_2, \Delta q = 0.5$ (g) $\ell_2, \Delta q = 1$ (h) $\ell_2, \Delta q = 1.5$

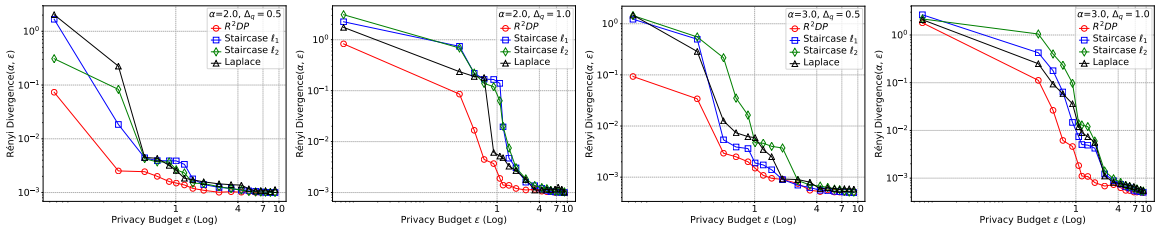
Figure 27: ℓ_1 and ℓ_2 metrics: R^{2}DP compared to Laplace and Staircase mechanisms for statistical queries (with five PDFs, i.e., Gamma, Uniform, Truncated Gaussian, Noncentral Chi-squared and Rayleigh distributions).



(a) $\Delta q = 0.1$ (b) $\Delta q = 0.5$ (c) $\Delta q = 1$ (d) $\Delta q = 1.5$

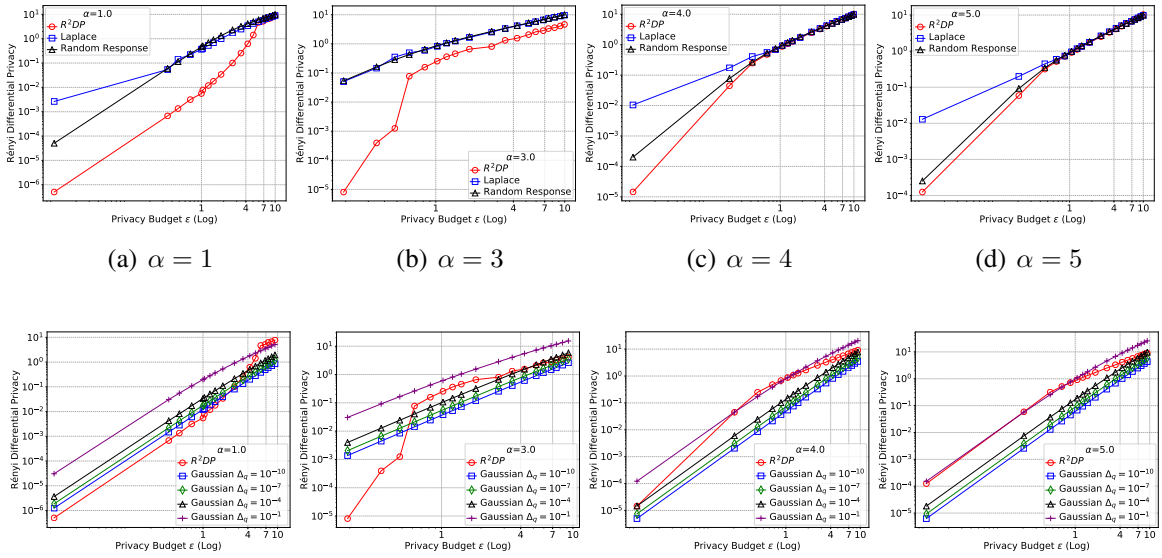
Figure 28: KL Divergence (Relative entropy metric): R^{2}DP (with five PDFs, i.e., Gamma, Uniform, Truncated Gaussian, Noncentral Chi-squared and Rayleigh distributions) compared to Laplace and Staircase mechanisms.

the improvement is the ratio between γ and Δq, since it exponentially affects the search space of the R^{2}DP mechanism. Furthermore, we observe that the Laplace and the staircase mechanisms are not optimal (w.r.t. usefulness) for very small and large values of ϵ, respectively, even though they are known to be optimal under other utility metrics (e.g., [74]).



(a) $\alpha = 2, \Delta q = 0.5$ (b) $\alpha = 2, \Delta q = 1$ (c) $\alpha = 3, \Delta q = 0.5$ (d) $\alpha = 2, \Delta q = 1$

Figure 29: Rényi Divergence (Relative entropy metric): R^{2}DP (with five PDFs, i.e., Gamma, Uniform, Truncated Gaussian, Noncentral Chi-squared and Rayleigh distributions) compared to Laplace and Staircase mechanisms.



(a) $\alpha = 1$ (b) $\alpha = 3$ (c) $\alpha = 4$ (d) $\alpha = 5$

(e) $\alpha = 1$ (f) $\alpha = 3$ (g) $\alpha = 4$ (h) $\alpha = 5$

Figure 30: Rényi Differential Privacy: (a-d) R^{2}DP compared to Laplace and Random Response mechanisms, and (e-h) R^{2}DP compared to Gaussian mechanism.

4.4.2.2 ℓ_1 and ℓ_2 Metrics

We compare R^{2}DP with the baseline Laplace and Staircase mechanisms [82], by varying the privacy budget ϵ and for four different sensitivities $\Delta q \in \{0.1, 0.5, 1, 1.5\}$. Our results validate the findings of Geng et al. [74], i.e., in the low privacy regime ($\epsilon \to \infty$), the Staircase mechanism is optimal while in the high privacy regime ($\epsilon \to 0$), the Laplace mechanism is optimal.

More importantly, our evaluations show that, for medium regime of privacy budgets (which

could be more desirable in practice), the class of optimal noise can be totally different. In fact, as shown in Geng et al. [74], the lower-bound of ϵ at which the Staircase distribution performs better than the Laplace distribution is somewhere around $\epsilon = 3$ for both ℓ_1 and ℓ_2 metrics. As illustrated in Figure 27, in contrast to ℓ_1 metric (for which the results of laplace and staircase are relatively tight), R^{2}DP can find a class of noises with significantly improved ℓ_2 metric for $\epsilon < 3$ (a logarithmic X axis is used to illustrate the performance in this region). The PDF of this class of noises is mostly two-fold distributions with Laplace distribution as the first fold, and Gamma distribution as the second fold. This finding is in line with the optimal class of noise proposed by Koufogiannis et al. [102], i.e., $f(v) = \frac{\epsilon^n \Gamma(\frac{n}{2}+1)}{\pi^{\frac{n}{2}}\Gamma(n+1)} e^{-\epsilon||v||_2}$. Furthermore, our results suggest different classes of optimal noises (than those found in the literature) for different parameters, sensitivity, ϵ and p (index of ℓ norm). In particular, a larger p tends to provide larger search spaces for R^{2}DP optimization, which results in further improved results for $\epsilon < 3$ (Figure 27 (a,b,e,f) vs. (c,d,g,h)).

4.4.2.3 Relative Entropy Metric

As Wang et al [166] has already shown that the output entropy of ϵ-DP randomization mechanisms is lower bounded by $1 - \ln(\epsilon/2)$ (for count queries) and the optimal result is achieved with Laplace mechanism, we focus our entropy metric evaluation on relative entropy metrics, i.e., KL and Rényi divergences. To define the prior distribution for this group of experiments, we have created a histogram with 50 bins of our data and calculated the probability mass function (pmf) of the bins. [1] As illustrated in Figure 28, we can draw similar observations for the KL entropy metric. In particular, we observe that R^{2}DP performs better for smaller sensitivity due to the larger search space of PDFs used in optimization. Similarly the Rényi entropy depicted in Figure 29 shows a similar trend with different α (the index of the divergence).

[1] 2 millions records fall into 50 bins (e.g., equal range for each bin). Then, any counting and moving average query (with different sensitivities) can be performed within each of the 50 bins to generate the distribution. Finally, the distance between the original and noisy distributions can be measured using the relative entropy metrics.

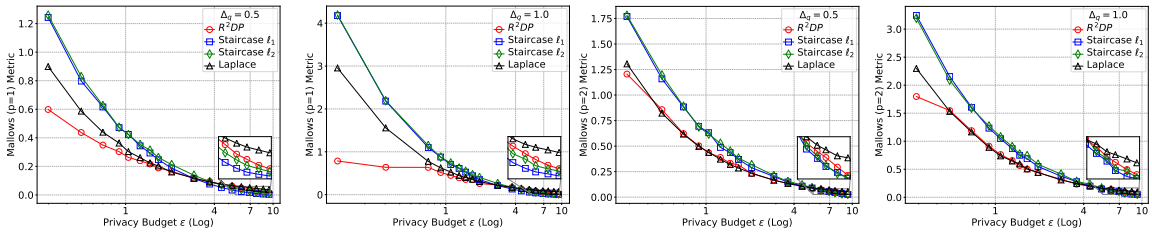
Summary. The R^{2}DP mechanism can generate better results than most of the well-known distributions for utility metrics without known optimal distributions (e.g., usefulness), and our results asymptotically approach to the optimal for utility metrics with known optimal distributions (e.g., ℓ_1 and ℓ_2). In particular, even though R^{2}DP is not specifically designed to optimize ℓ_1 and ℓ_2 metrics, we observe very similar performance between the R^{2}DP results and the optimal Staircase results, e.g., the multiplicative gain compared to the Laplace results. We note that using a larger number of independent RVs drawn from different PDFs as the search space generator may further improve the results.

4.4.3 Tightness of R^{2}DP under Rényi DP

Rényi differential privacy [129] is a recently proposed as a relaxed notion of DP which effectively quantifies the bad outcomes in (ϵ, δ)-DP mechanisms and consequently evaluates how such mechanisms behave under sequential compositions (see Appendix 4.5.8.2 for details on Rényi DP). We now evaluate how the privacy loss of R^{2}DP behaves under Rényi DP.

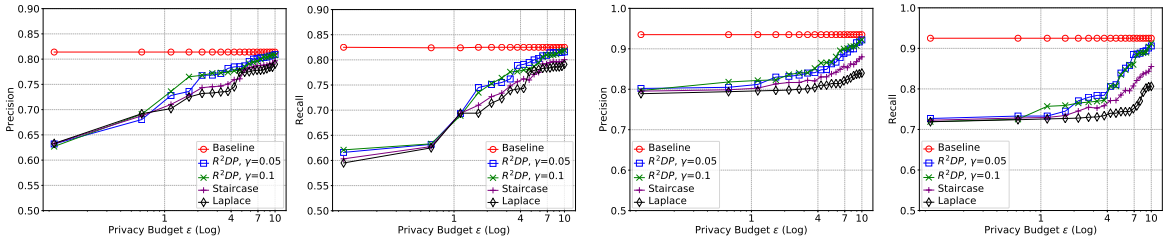
Specifically, this group of experiments are conducted to provide insights about the privacy loss of R^{2}DP and other well-known mechanisms. In particular, Figure 30 (a-d) depicts the Rényi differential privacy of the R^{2}DP and two basic mechanisms for counting queries: random response and Laplace mechanisms. These results are based on the privacy guarantees depicted in Table 7. Our results demonstrate that fine tuning R^{2}DP can generate strictly more private results compared to the other two ϵ-DP mechanisms when the definition of the privacy notion is relaxed. Furthermore, the level of such tightness depends on the Rényi differential privacy index where a smaller value of α pertains to a relatively tighter R^{2}DP mechanism. On the other hand, all three mechanisms behave more similarly as α increases. Ultimately, at $\alpha \to \infty$, where Rényi differential privacy becomes equivalent to the classic notion of ϵ-DP, all three mechanisms' privacy guarantees converge to ϵ.

In the next set of experiments, we compare the R^{2}DP mechanism and Gaussian mechanism in terms of privacy guarantee to understand how exactly the bad outcomes probability (δ) affects the



(a) $p = 1, \Delta q = 0.5$ (b) $p = 1, \Delta q = 1$ (c) $p = 2, \Delta q = 0.5$ (d) $p = 2, \Delta q = 1$

Figure 31: Mallows metric: R^{2}DP compared to Laplace and Staircase mechanisms for degree distribution (Facebook dataset).



(a) Precision vs. ϵ (UCI Adult) (b) Recall vs. ϵ (UCI Adult) (c) Precision vs. ϵ (KDDCup99) (d) Recall vs. ϵ (KDDCup99)

Figure 32: Accuracy evaluation for classification (UCI Adult dataset and KDDCup99 dataset)

Table 7: Summary of Rényi DP parameters for four mechanisms based on Theorem 4.5.13

Mechanism	**Differential Privacy**	**Rényi Differential Privacy for α**
Laplace	$\frac{1}{b}$	$\alpha > 1 : \frac{1}{\alpha-1} \log \left[\frac{\alpha \cdot exp(\frac{\alpha-1}{b}) + (\alpha-1) \cdot exp(\frac{-\alpha}{b})}{2\alpha-1} \right]$ $\alpha = 1 : \frac{1}{b} + exp(\frac{-1}{b}) - 1$
Random Response	$\lvert \log \frac{p}{1-p} \rvert$	$\alpha > 1 : \frac{1}{\alpha-1} \log [p^\alpha (1-p)^{1-\alpha} + p^{1-\alpha}(1-p)^\alpha]$ $\alpha = 1 : (2p-1) \log \frac{p}{1-p}$
R^{2}DP	$M'_{\frac{1}{b}}(0) / M'_{\frac{1}{b}}(-1)$	$\alpha > 1 : \frac{1}{\alpha-1} \log \left[\frac{\alpha M_{\frac{1}{b}}(\alpha-1) + (\alpha-1) M_{\frac{1}{b}}(-\alpha)}{2\alpha-1} \right]$ $\alpha = 1 : M'_{\frac{1}{b}}(0) + M_{\frac{1}{b}}(-1) - 1$
Gaussian	∞	$\frac{\alpha}{2\sigma^2}$

privacy robustness of a privatized mechanism. Figure 30 (e-h) gives such a comparison. Specifically, since Rényi differential privacy at each α can be seen as higher-order moments as a way of bounding the tails of the privacy loss variable [129], we observe that each value of α reveals a snapshot of such a privacy loss. As a tangible observation, we conclude that the class of optimal ϵ-differential privacy mechanisms benefits from a very smaller privacy loss at smaller moments (which are more decisive in overall protection) and larger privacy loss at bigger moments.

4.4.4 Social Network Analysis

We conduct experiments to compare the performance of R^{2}DP, Laplace and two staircase mechanisms based on PINQ queries in social network analysis. Figure 31 compares the degree distribution for a real Facebook dataset using Mallows metric (the prior, i.e.,$n = 47,538$ nodes, and $p = 1$ or 2 for computing the distribution distance using Mallows metric). Again, our results confirm that R^{2}DP can effectively generate PDFs to maximize this utility metric suitable for social networking analysis. Note that, since the definition of this metric is similar to ℓ_p metric (Mallows is more empirical, depending on the number of nodes in the dataset), the results for this metric display a similar pattern to those for ℓ_p metric depicted in Figure 27.

4.4.5 Machine Learning

We obtain our baseline results by applying the Naive Bayes classifier on the Adult dataset (45K training records and 5K testing records), the precision and recall results are derived as 0.814 and 0.825, respectively. Then, we evaluate the precision and recall of R^{2}DP and Laplace-based naive classification [162] by varying the privacy budget for each PINQ query $\epsilon \in [0.1, 10]$ (sensitivity=1) where two different error bounds $\gamma = 0.05, 0.1$ are specified for R^{2}DP. We have the following observations:

- As shown in Figure 32(a) and 32(b), the R^{2}DP-based classification is more accurate than the Laplace and staircase mechanisms with the same total privacy budget for all the PINQ queries ϵ. As the privacy budget ϵ increases, following our statistical query experiments, R^{2}DP offers a far better precision/recall compared to the Laplace-based classification (close to the results without privacy consideration) since it approaches to the optimal PDF.

- Among the precision/recall results derived with two different γ in R^{2}DP-based classification, for each ϵ, one out of the two specified error bounds (e.g., $\gamma = 5\%$) may reach the highest accuracy (not necessarily the result with the smaller γ).

- As shown in Figure 32(c) and 32(d), we can draw similar observations from the KDDCup99 dataset.

The above experimental results have validated the effectiveness of integrating R^{2}DP to improve the output utility for classification while ensuring ϵ-differential privacy. In summary, all the experiments conducted in both statistical queries and real-world applications have validated the practicality of the R^{2}DP framework.

4.5 Proofs and Further Discussions

4.5.1 Demonstration of Theorem 4.2.1

A Laplace distribution is of a $(\propto x \cdot e^{x \cdot t})$ order, where x is the inverse of the scale parameter. Second, since $x \cdot e^{x \cdot t} = \dfrac{\mathrm{d}e^{x \cdot t}}{\mathrm{d}t}$, the cumulative distribution function (CDF) resulted from randomizing x can be expressed in terms of the expectation $\mathbb{E}(e^{x \cdot t})$. We note that from now on, we will simply refer to R^{2}DP with Laplace distribution as the first fold PDF as *the R^{2}DP mechanism*.

Example 4.5.1. *Following Example 4.2.1, for a Bernoulli distributed scale parameter b, Figure 33 illustrates the above finding (see Appendix 4.5.3 for proof). It can be verified that the term inside the braces is the derivative of $\mathbb{E}(e^{\frac{1}{b} \cdot -|w|})$ w.r.t. $-|w|$, and hence the above probability can be expressed in terms of the expectation.*

4.5.2 Case Study PDFs

4.5.2.1 Discrete Probability Distributions

First, we consider two different mixture Laplace distributions that can be applied for constructing R^{2}DP with discrete probability distribution f_b.

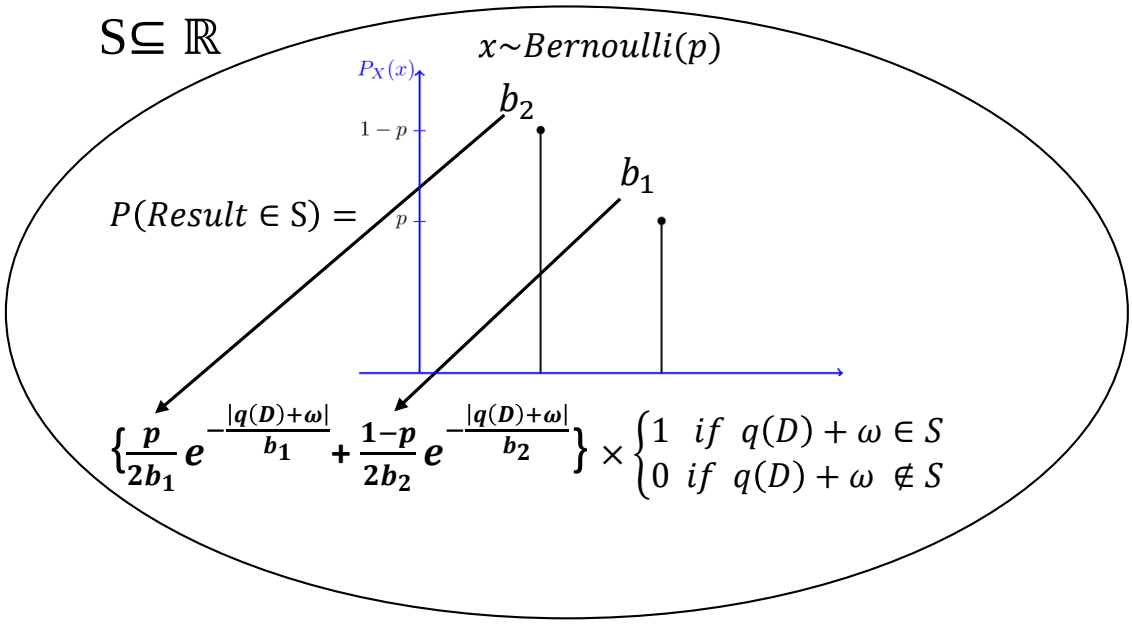


Figure 33: The term in the parenthesis is the derivative of $\mathbb{E}(e^{\frac{1}{b}\cdot-|w|})$ w.r.t. $-|w|$, and hence the above probability can be expressed in terms of the expectation

(1) **Degenerate distribution.** A degenerate distribution is a probability distribution in a (discrete or continuous) space with support only in a space of lower dimension [25]. If the degenerate distribution is uni-variate (involving only a single random variable), it will be a deterministic distribution and takes only a single value. Therefore, the degenerate distribution is identical to the baseline Laplace mechanism as it also assigns the mechanism one single scale parameter b_0. Specifically, the probability mass function of the uni-variate degenerate distribution is:

$$f_{\delta,k_0}(x) = \begin{cases} 1 & x = k_0 \\ 0 & x \neq k_0 \end{cases}$$

The MGF for the degenerate distribution δ_{k_0} is given by $M_k(t) = e^{t \cdot k_0}$ [28]. Using Equation 9, Theorem 4.5.1 gives the same DP guarantee as the baseline Laplace mechanism.

Theorem 4.5.1. *The R^2DP mechanism $M_q(d, \epsilon)$, $\epsilon \sim f_{\delta,\frac{1}{b_0}}(\epsilon)$, is $\frac{\Delta q}{b_0}$-differentially private.*

Obviously, this distribution does not improve the bound in Theorem 4.3.2 but shows the soundness of our findings.

(2) **Bernoulli distribution.** The probability mass function of this distribution, over possible outcomes k, is

$$f_B(k;p) = \begin{cases} p & \text{if } k = 1, \\ 1-p & \text{if } k = 0. \end{cases}$$

Note that the binary outcomes $k = 0$ and $k = 1$ can be mapped to any two outcomes X_0 and X_1, respectively. Therefore, we consider the following Bernoulli outcomes

$$f_{B,X_0,X_1}(X;p) = \begin{cases} p & \text{if } X = X_1, \\ 1-p & \text{if } X = X_0. \end{cases}$$

The MGF for Bernoulli distribution $f_{B,X_0,X_1}(X;p)$ is $M_X(t) = p\cdot e^{t\cdot X_0} + (1-p)\cdot e^{t\cdot X_1}$ [28]. We now derive the precise differential privacy guarantee of an R^{2}DP mechanism with its scale parameter randomized according to a Bernoulli distribution.

Theorem 4.5.2. *The R^{2}DP mechanism $M_q(d,\epsilon)$, $\epsilon \sim f_{B,\frac{1}{b_0},\frac{1}{b_1}}(\epsilon;p)$, satisfies $\ln[p\cdot e^{\frac{\Delta q}{b_0}} + (1-p)\cdot e^{\frac{\Delta q}{b_1}}]$ differential privacy.*

This bound is exactly the mean of $e^{\epsilon(b)}$ given in Theorem 4.3.2.

4.5.2.2 Continuous Probability Distributions

We now investigate three compound Laplace distributions.

(1) **Gamma distribution.** The gamma distribution is a two-parameter family of continuous probability distributions with a shape parameter $k > 0$ and a scale parameter θ. Besides the generality, the gamma distribution is the maximum entropy probability distribution (both w.r.t. a uniform base measure and w.r.t. a $1/x$ base measure) for a random variable X for which $\mathbb{E}(X) = k\theta = \alpha/\beta$ is fixed and greater than zero, and $\mathbb{E}[\ln(X)] = \psi(k) + \ln(\theta) = \psi(\alpha) - \ln(\beta)$ is fixed (ψ is the

digamma function). Therefore, it may provide a relatively higher privacy-utility trade-off in comparison to the other candidates [99, 92]. A random variable X that is gamma-distributed with shape α and rate β is denoted by $X \sim \Gamma(k, \theta)$ and the corresponding PDF is

$$f_\Gamma(X; k, \theta) = \frac{x^{k-1} e^{-\frac{x}{\theta}}}{\Gamma(k) \cdot \theta^k} \quad \text{for } X > 0 \text{ and } k, \theta > 0,$$

where $\Gamma(\alpha)$ is the gamma function. We now investigate the differential privacy guarantee provided by assuming that the reciprocal of the scale parameter b in Laplace mechanism is distributed according to the gamma distribution (see Appendix 4.5.3 for the proof).

Theorem 4.5.3. *The R^{2}DP mechanism $M_q(d, \epsilon)$, $\epsilon \sim f_\Gamma(\epsilon; k, \theta)$, satisfies $\big((k+1) \cdot \ln(1 + \Delta q \cdot \theta)\big)$ differential privacy.*

We now apply the necessary condition given in Equation 13 (see Appendix 4.5.3 for the proof).

Lemma 4.5.4. *R^{2}DP using Gamma distribution can satisfy the necessary condition in Equation 13.*

Therefore, Gamma distribution may improve over the baseline, and this can be computed by optimizing the privacy-utility trade-off using the Lagrange multiplier function in Equation 11. Also, our numerical results show that, this distribution is more effective for large ϵ (weaker privacy guarantees).

(2) **Uniform distribution.** In probability theory and statistics, the continuous uniform distribution or rectangular distribution is a family of symmetric probability distributions such that for each member of the family, all intervals of the same length on the support of the distribution are equally probable. The support is defined by the two parameters, a and b, which are the minimum and maximum values. The distribution is often abbreviated as $U(a, b)$, which is the maximum entropy probability distribution for a random variable X under no constraint; other than that, it is contained

in the distribution's support [99, 92]. The MGF for $U(a, b)$ is

$$M_X(t) = \begin{cases} \frac{e^{tb}-e^{ta}}{t(b-a)} & \text{for } t \neq 0, \\ 1 & \text{for for } t = 0. \end{cases}$$

Using Theorem 4.3.1, we now drive the precise differential privacy guarantee of an R^{2}DP mechanism for uniform distribution $U(a, b)$.

Theorem 4.5.5. *The R^{2}DP mechanism $M_q(d, \epsilon)$, $\epsilon \sim f_{U(a,b)}(\epsilon)$, is $\ln\left[\frac{\alpha^2-\beta^2}{2((1+\beta)e^{-\beta}-(1+\alpha)e^{-\alpha})}\right]$-differentially private, where $\alpha = a \cdot \Delta q$ and $\beta = b \cdot \Delta q$.*

We now apply the necessary condition given in Equation 13. One can easily verify that the inequality holds for an infinite number of settings, e.g., $a = 0.5$, $b = 9$ and $\Delta q = 1.2$.

Lemma 4.5.6. *R^{2}DP using uniform distribution can satisfy the necessary condition in Equation 13.*

Therefore, R^{2}DP using uniform distribution may improve over the baseline, and this can be computed by optimizing the privacy-utility trade-off using the Lagrange multiplier function in Equation 11. Also, our numerical results show that, this distribution can also be effective for both small and large ϵ.

(3) **Truncated Gaussian distribution.** The last distribution we consider is the Truncated Gaussian distribution. This distribution is derived from that of a normally distributed random variable by bounding the random variable from either below or above (or both). Therefore, we can benefit from the numerous useful properties of Gaussian distribution, by truncating the negative region of the Gaussian distribution. Suppose $X \sim \mathcal{N}(\mu, \sigma^2)$ has a Gaussian distribution and lies within the interval $X \in (a, b),\ -\infty \leq a < b \leq \infty$. Then, X conditional on $a < X < b$ has a truncated Gaussian distribution with the following probability density function

$$f_{\mathcal{N}^T}(X; \mu, \sigma, a, b) = \frac{\phi(\frac{X-\mu}{\sigma})}{\sigma \cdot \left(\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})\right)} \quad \text{for } a \leq x \leq b$$

and by $f_{\mathcal{N}^T} = 0$ otherwise. Here, $\phi(x) = \frac{1}{\sqrt{2\pi}.} e^{-\frac{x^2}{2}}$ and $\Phi(x) = 1 - Q(x)$ are PDF and CDF of the standard Gaussian distribution, respectively. Next, using Theorem 4.3.1, we give the differential privacy guarantee provided by the mechanism assuming that the reciprocal of b is distributed according to the truncated Gaussian distribution.

Theorem 4.5.7. *The R^{2}DP mechanism $\mathcal{M}_q(d, \epsilon)$, $\epsilon \sim f_{\mathcal{N}^T}(\epsilon; \mu, \sigma, a, b)$, satisfies ϵ_{N^T}- differential privacy, where*

$$\epsilon_{N^T} = \ln \left[\frac{\mu + \dfrac{\sigma \cdot (\phi(\alpha) - \phi(\beta))}{(\Phi(\beta) - \Phi(\alpha))}}{\dfrac{\mathrm{d}M_{N^T}(t)}{\mathrm{d}t}|_t = -\Delta q} \right] \tag{14}$$

in which $\phi(\cdot)$ is the probability density function of the standard normal distribution, $\phi(\cdot)$ is its cumulative distribution function and $\alpha = \frac{a-\mu}{\sigma}$ and $\beta = \frac{b-\mu}{\sigma}$.

Lemma 4.5.8 (see Appendix 4.5.3 for the proof)**.** *R^{2}DP using truncated Gaussian distribution can satisfy the necessary condition in Equation 13.*

Therefore, truncated Gaussian distribution may improve over the baseline, and this can be computed by optimizing the privacy-utility trade-off using the Lagrange multiplier function in Equation 11. In particular, our numerical results show that, this distribution can also be effective for smaller ϵ (stronger privacy guarantees).

4.5.3 Proofs

Example 4.5.1. Following Example 4.2.1, for a Bernoulli distributed scale parameter b, we have

$$\begin{aligned}
&\mathbb{P}(\mathcal{M}_q(d,b) \in S) \\
&= \int_{\mathbb{R}} \frac{p}{2b_1} \cdot \mathbb{1}_S\{q(d)+w\} e^{\frac{-|w|}{b_1}} + \frac{1-p}{2b_2} \cdot \mathbb{1}_S\{q(d)+w\} e^{\frac{-|w|}{b_2}} dw \\
&= \int_{\mathbb{R}} \big(\frac{p}{2b_1} \cdot e^{\frac{-|w|}{b_1}} + \frac{1-p}{2b_2} \cdot e^{\frac{-|w|}{b_2}}\big) \mathbb{1}_S\{q(d)+w\} dw
\end{aligned}$$

where $\mathbb{1}_{\{\cdot\}}$ denotes the indicator function. It can be verified that the term in the braces is the derivative of $\mathbb{E}(e^{\frac{1}{b}\cdot -|w|})$ w.r.t. $-|w|$, and hence the above probability can be expressed in terms of the expectation. □

Theorem 4.2.1. For an R^{2}DP Laplace mechanism and $\forall S \subset \mathbb{R}$ measurable and dataset d in D, we have

$$\begin{aligned}
&\mathbb{P}(\mathcal{M}_q(d,b) \in S) \\
&= \int_{\mathbb{R}_{\geq 0}} f(b) \frac{1}{2b} \int_{\mathbb{R}} \mathbb{1}_S\{q(d)+w\} e^{\frac{-|w|}{b}} \, dw \, db \\
&= \int_{\mathbb{R}_{\geq 0}} g(u) \frac{u}{2} \int_{\mathbb{R}} \mathbb{1}_S\{q(d)+w\} e^{-|w|\cdot u} \, dw \, du \\
&= \int_{\mathbb{R}} \mathbb{1}_S\{q(d)+w\} \int_{\mathbb{R}_{\geq 0}} g(u) \frac{u}{2} e^{-|w|\cdot u} \, du \, dw \\
&= \int_{\mathbb{R}} \mathbb{1}_S\{q(d)+w\} \frac{1}{2} \frac{dM_u(t)}{dt}|_{t=-|w|} \, dw \\
&= \tfrac{1}{2} \int_S \frac{dM_u(t)}{dt}|_{t=-|x-q(d)|} dx
\end{aligned} \tag{15}$$

$$= \tfrac{1}{2} \cdot \Big[- M_u(-|x-q(d)|)|_{S_{\geq q(d)}} + M_u(-|x-q(d)|)|_{S_{<q(d)}} \Big] \tag{16}$$

where $u = b^{-1}$, is reciprocal of random variable b and $g(u) = \frac{1}{u^2} \cdot f(\frac{1}{u})$. Note that $M_u(t)$ is the

MGF of random variable u which is identical with $M_{\frac{1}{b}}(t)$. $\square$

Theorem 4.3.1. To prove this theorem, we first need to give two lemmas on the properties of R^2DP Laplace mechanism and MGFs.

Lemma 4.5.9. *The R^2DP mechanism $\mathcal{M}_q(d, b)$, is*

$$\ln\left[\max_{\forall x\in\mathbb{R}}\left\{\frac{\frac{\mathrm{d}M_{\frac{1}{b}}(t)}{\mathrm{d}t}|_{t=-|x-q(d)|}}{\frac{\mathrm{d}M_{\frac{1}{b}}(t)}{\mathrm{d}t}|_{t=-|x-q(d')|}}\right\}\right] \text{ -differentially private.} \tag{17}$$

Proof. According to Equation 15,

$$\begin{aligned}\mathbb{P}(\mathcal{M}_q(d,b)\in S) &= \tfrac{1}{2}\int_S \frac{\mathrm{d}M_{\frac{1}{b}}(t)}{\mathrm{d}t}|_{t=-|x-q(d)|}dx \\ &= \tfrac{1}{2}\int_S \frac{\frac{\mathrm{d}M_{\frac{1}{b}}(t)}{\mathrm{d}t}|_{t=-|x-q(d)|}}{\frac{\mathrm{d}M_{\frac{1}{b}}(t)}{\mathrm{d}t}|_{t=-|x-q(d')|}}\cdot\frac{\mathrm{d}M_{\frac{1}{b}}(t)}{\mathrm{d}t}|_{t=-|x-q(d')|}dx\end{aligned}$$

Denote by

$$\begin{aligned}e^{\epsilon} &= \sup\left\{\frac{\frac{\mathrm{d}M_{\frac{1}{b}}(t)}{\mathrm{d}t}|_{t=-|x-q(d)|}}{\frac{\mathrm{d}M_{\frac{1}{b}}(t)}{\mathrm{d}t}|_{t=-|x-q(d')|}}, \forall x\in S\right\}, \\ \Rightarrow \mathbb{P}(\mathcal{M}_q(d,b)\in S) &\le e^{\epsilon}\cdot\mathbb{P}(\mathcal{M}_q(d',b)\in S)\end{aligned}$$

and the choice of $S=\mathbb{R}$ concludes the proof. $\square$

Next, we show the log-convexity property of the first derivative of moment generating functions.

Lemma 4.5.10. *First derivative of a moment generating function defined by $\frac{\mathrm{d}M(t)}{\mathrm{d}t} = \mathbb{E}(z\cdot e^{zt})$ is log-convex.*

Proof. For real- or complex-valued random variables X and Y, Hölder's inequality [1] reads; $\mathbb{E}(|XY|) \le (\mathbb{E}(|X|)^p)^{1/p}\cdot(\mathbb{E}(|Y|)^q)^{1/q}$ for any $1 < p, q < \infty$ with $1/p + 1/q = 1$. Next, for

all $\theta \in (0,1)$ and $0 \leq x_1, x_2 < \infty$, define $X = z^{\theta} \cdot e^{\theta x_1 z}$, $Y = z^{1-\theta} \cdot e^{(1-\theta)x_2 z}$ and $p = 1/\theta$, $q = 1/(1-\theta)$. Therefore, we have

$$\mathbb{E}(z \cdot e^{(\theta x_1 + (1-\theta)x_2 z}) \leq \mathbb{E}(z \cdot e^{x_1 z})^{\theta} \cdot \mathbb{E}(z \cdot e^{x_2 z})^{1-\theta}$$

which shows the definition of log-convexity holds for $M'(t)$. □

Back to the original proof, following the DP guarantee in Lemma 4.5.9, and using triangle inequality, we have

$$e^{\epsilon} = \max_{\forall x \in \mathbb{R}} \left\{ \frac{\mathbb{E}(\epsilon \cdot e^{(-|x-q(d)|\cdot\epsilon)})}{\mathbb{E}(\epsilon \cdot e^{(-|x-q(d')|\cdot\epsilon)})} \right\} \leq \max_{\forall t \in \mathbb{R}_{\leq 0}} \left\{ \frac{\mathbb{E}(\epsilon \cdot e^{(t\cdot\epsilon)})}{\mathbb{E}(\epsilon \cdot e^{((t-\Delta q)\cdot\epsilon)})} \right\}$$

Next, we show that $f(t) = \frac{\mathbb{E}(\epsilon \cdot e^{(t\cdot\epsilon)})}{\mathbb{E}(\epsilon \cdot e^{((t-\Delta q)\cdot\epsilon)})}$ is non-decreasing w.r.t. t. For this purpose, we must show that

$$f'(t) = \frac{M''(t) \cdot M'(t-\Delta q) - M'(t) \cdot M''(t-\Delta q)}{M'^2(t-\Delta q)}$$

is non-negative. However, this is equivalent to show that $\frac{M''(t)}{M'(t)} \geq \frac{M''(t-\Delta q)}{M'(t-\Delta q)}$ or more generally $\frac{M''(t)}{M'(t)}$ is not-decreasing. However, following the log-convexity of first $M'(t)$, the logarithmic derivative of $M'(t)$ denoted by $\frac{M''(t)}{M'(t)}$ is non-decreasing. Thus, for all $t < 0$, $f(t) \leq f(0)$, and evaluating $e^{\epsilon(t)}$ at $t = 0$, concludes our proof. □

Theorem 4.3.2. Following Theorem 1.5.1, an ϵ-DP Laplace mechanism is $(\gamma, e^{\frac{-\gamma}{b(\epsilon)}})$-useful for all $\gamma \geq 0$, where $b(\epsilon) = \frac{\Delta q}{\epsilon}$. Therefore, for the usefulness of the baseline Laplace mechanism at $\epsilon = \ln[\mathbb{E}_{\frac{1}{b}}(e^{\epsilon(b)})]$, we have

$$e^{\frac{-\gamma \cdot \ln[\mathbb{E}_{\frac{1}{b}}(e^{\epsilon(b)})]}{\Delta q}} = \left(\mathbb{E}_{\frac{1}{b}}(e^{\epsilon(b)})\right)^{\frac{-\gamma}{\Delta q}} = \left(\mathbb{E}_{\frac{1}{b}}(e^{\frac{\Delta q}{b}})\right)^{\frac{-\gamma}{\Delta q}} \leq \mathbb{E}_{\frac{1}{b}}\left(e^{\frac{-\gamma}{b}}\right)$$

where the last inequality relation is verified by Jensen inequality [93] as $g(x) = x^{\frac{-\gamma}{b}}$ is a convex function. Recall the following Jensen inequality: Let $(\Omega, \mathfrak{F}, \mathrm{P})$ be a probability space, X an integrable real-valued random variable and g a convex function. Then

$$g(\mathbb{E}(X)) \leq \mathbb{E}(g(X))$$

Therefore,

$$1 - e^{\frac{-\gamma \cdot \ln[\mathbb{E}_{\frac{1}{b}}(e^{\epsilon(b)})]}{\Delta q}} \geq 1 - \mathbb{E}_{\frac{1}{b}}\left(e^{\frac{-\gamma}{b}}\right) = U(\ln[\mathbb{E}_{\frac{1}{b}}(e^{\epsilon(b)})], \Delta q, \gamma)$$

This completes the proof. □

Theorem 4.5.1. For $\frac{1}{b} \sim f_{\delta, \frac{1}{b_0}}(\frac{1}{b})$, the MGF is given by $M_{\frac{1}{b}}(t) = e^{\frac{t}{b_0}}$. Following Theorem 4.5.9, one can write

$$\begin{aligned} e^{\epsilon} &= \max_{\forall x \in \mathbb{R}} \left\{ \frac{\frac{1}{b_0} \cdot e^{\frac{-|x-q(d)|}{b_0}}}{\frac{1}{b_0} \cdot e^{\frac{-|x-q(d')|}{b_0}}} \right\} = \max_{\forall x \in \mathbb{R}} \left\{ e^{\frac{|x-q(d')|-|x-q(d)|}{b_0}} \right\} \\ &\leq \max_{\forall x \in \mathbb{R}} \left\{ e^{\frac{|q(d)-q(d')|}{b_0}} \right\} = e^{\frac{\Delta q}{b_0}} \end{aligned}$$

where the last inequality is from triangle inequality.

□

Theorem 4.5.2. The R^{2}DP Laplace mechanism $\mathcal{M}_q(d, b)$, $\frac{1}{b} \sim f_{B, \frac{1}{b_0}, \frac{1}{b_1}}(\frac{1}{b}; p)$ returns with probability p, a Laplace mechanism with scale parameter b_1, and with probability $1 - p$ another Laplace mechanism with scale parameter b_2. To this end, we are looking for

$$e^{\epsilon} = \max_{\forall x \in \mathbb{R}} \left\{ \frac{\frac{p}{b_0} \cdot e^{\frac{-|x-q(d)|}{b_0}} + \frac{1-p}{b_1} \cdot e^{\frac{-|x-q(d)|}{b_1}}}{\frac{p}{b_0} \cdot e^{\frac{-|x-q(d')|}{b_0}} + \frac{1-p}{b_1} \cdot e^{\frac{-|x-q(d')|}{b_1}}} \right\}$$

Therefore, using triangle inequality, we have

$$e^{\epsilon_1} = \max_{\forall S \in \mathbb{R}} \left\{ \frac{p \cdot e^{\frac{-|x-q(d)|}{b_0}} + (1-p) \cdot e^{\frac{-|x-q(d)|}{b_1}}}{p \cdot e^{\frac{-|x-q(d')|}{b_0}} + (1-p) \cdot e^{\frac{-|x-q(d')|}{b_1}}} \right\}$$
$$\leq \max_{\forall x \geq q(d)} \left\{ \frac{p \cdot e^{\frac{\Delta q - |x-q(d')|}{b_0}} + (1-p) \cdot e^{\frac{\Delta q + -|x-q(d')|}{b_1}}}{p \cdot e^{\frac{-|x-q(d')|}{b_0}} + (1-p) \cdot e^{\frac{-|x-q(d')|}{b_1}}} \right\}$$

Let us make the substitutions $X = e^{\frac{-|x-q(d')|}{b_0}}$, $a = e^{\frac{\Delta q}{b_0}}$ and k=$\frac{b_0}{b_1} > 1$. Hence, we have

$$e^{\epsilon} \leq \max_{\forall X \in (0,1)} \left\{ \frac{p \cdot a \cdot X + (1-p) \cdot (a \cdot X)^k}{p \cdot X + (1-p) \cdot X^k} \right\}$$

To obtain e^{ϵ}, we need to find all the critical points of $e^{\epsilon_1}(X) = \frac{p \cdot a \cdot X + (1-p) \cdot (a \cdot X)^k}{p \cdot X + (1-p) \cdot X^k}$. However, the critical points of a fractional function are the roots of the numerator of its derivative. Hence, suppose

$$\frac{\mathrm{d}e^{\epsilon}(X)}{\mathrm{d}X} = \frac{N(X)}{D(X)}$$

then

$$\Rightarrow N(X) = \big(p \cdot a + (1-p) \cdot k \cdot a \cdot (a \cdot X)^{k-1}\big)$$
$$\cdot \big(p \cdot X + (1-p) \cdot X^k\big) - \big(p + (1-p) \cdot k \cdot X^{k-1}\big)$$
$$\cdot \big(p \cdot a \cdot X + (1-p) \cdot (a \cdot X)^k\big)$$
$$= p \cdot (1-p) \cdot (k-1) \cdot (a^{k-1} - 1) \cdot X^k$$

However, all the terms in the last expression are strictly positive. Therefore, the only critical points are $X = 0$ and $X = 1$ and as the function is strictly increasing,

$$e^{\epsilon} \leq e^{\epsilon}(1) = p \cdot a + (1-p) \cdot (a)^k$$
$$= p \cdot e^{\frac{\Delta q}{b_0}} + (1-p) \cdot e^{\frac{\Delta q}{b_1}}$$

which is the bound in the Theorem. □

Theorem 4.5.3. For a Gamma distribution with shape parameters k and scale parameters θ, the MGF at point t is given as $(1-\theta\cdot t)^{-k}$. Since $\frac{1}{b}\sim f_{\Gamma}(\frac{1}{b};k,\theta)$, following Theorem 4.5.9, one can write

$$e^{\epsilon}=\max_{\forall x\in\mathbb{R}}\left\{\frac{k\cdot\theta\cdot(1+\theta\cdot|x-q(d)|)^{-k-1}}{k\cdot\theta\cdot(1+\theta\cdot|x-q(d')|)^{-k-1}}\right\}$$
$$\Rightarrow\epsilon=\max_{\forall x\in\mathbb{R}}\left\{(k+1)\cdot\ln\left[\frac{(1+\theta\cdot|x-q(d')|)}{(1+\theta\cdot|x-q(d)|)}\right]\right\}$$

to find the maximum of the $\ln$ term, denote by $X=1+\theta\cdot|x-q(d)|$). Moreover, since $|x-q(d')|\leq|x-q(d)|+\Delta q$, we have

$$\Rightarrow\epsilon\leq\max_{\forall X\geq 1}\left\{\frac{X+\Delta q\cdot\theta}{X}\right\}$$

However, since

$$\forall X\geq 1,\ \frac{X+\Delta q\cdot\theta}{X}$$

is strictly decreasing, we have

$$\Rightarrow\epsilon=(k+1)\cdot\ln\left[1+\theta\cdot\Delta q\right]$$

This completes the proof. □

Lemma 4.5.4. We need to show that there exist k and θ such that $(k+1)\cdot\ln(1+\Delta q\cdot\theta)<-k\cdot\ln(1-\Delta q\cdot\theta)$, $\theta<\frac{1}{\Delta q}$. Given $\theta=\frac{1}{2\Delta q}$, we need to show that $\exists k, k\cdot\ln(2)>(k+1)\cdot\ln(1.5)$, which always holds for all $k>1.4094$. □

Lemma 4.5.8. Using exhaustive search, suppose $\mu = 0.5223$,$\sigma = 1.5454$, $a = 0.5223$ and for $\epsilon = 1.1703$ and $\Delta q = 0.6$, we will get $\ln(M_{\mathcal{N}^T}(\Delta q)) = 1.2417$. □

4.5.4 Lagrange Multiplier Function

The Lagrange Multiplier Function (all possible linear combinations of the Gamma, uniform and truncated Gaussian distributions) is:

$$\begin{aligned}
&\mathcal{L}(a_1, a_2, a_3, k, \theta, a_u, b_u, \mu, \sigma, a_{\mathcal{N}^T}, b_{\mathcal{N}^T}, \Lambda) \qquad (18)\\
&= M_{\Gamma(k,\theta)}(-a_1\gamma) \cdot M_{U(a_u,b_u)}(-a_2\gamma)\\
&\cdot M_{\mathcal{N}^T(\mu,\sigma,a_{\mathcal{N}^T},b_{\mathcal{N}^T})}(-a_3\gamma) + \Lambda \cdot (\ln\left[\frac{\mathsf{N}}{\mathsf{D}}\right] - \epsilon)
\end{aligned}$$

where the numerator and the denominator $\mathsf{N},\ \mathsf{D}$ are

$$\mathsf{N} = (a_1 \cdot k \cdot \theta) + (a_2 \cdot \tfrac{a+b}{2}) + (a_3 \cdot (\mu + (\frac{\sigma \cdot \phi(\alpha) - \phi(\beta))}{(\Phi(\beta) - \Phi(\alpha))}))$$

$$\begin{aligned}
\mathsf{D} = a_1 \cdot M'_{\Gamma(k,\theta)}(-a_1 \cdot \Delta q) \cdot M_{U(a_u,b_u)}(-a_2 \cdot \Delta q)\\
\cdot M_{\mathcal{N}^T(\mu,\sigma,a_{\mathcal{N}^T},b_{\mathcal{N}^T})}(-a_3 \cdot \Delta q)\\
+a_2 \cdot M_{\Gamma(k,\theta)}(-a_1 \cdot \Delta q) \cdot M'_{U(a_u,b_u)}(-a_2 \cdot \Delta q)\\
\cdot M_{\mathcal{N}^T(\mu,\sigma,a_{\mathcal{N}^T},b_{\mathcal{N}^T})}(-a_3 \cdot \Delta q)\\
+a_3 \cdot M_{\Gamma(k,\theta)}(-a_1 \cdot \Delta q) \cdot M_{U(a_u,b_u)}(-a_2 \cdot \Delta q)\\
\cdot M'_{\mathcal{N}^T(\mu,\sigma,a_{\mathcal{N}^T},b_{\mathcal{N}^T})}(-a_3 \cdot \Delta q)
\end{aligned}$$

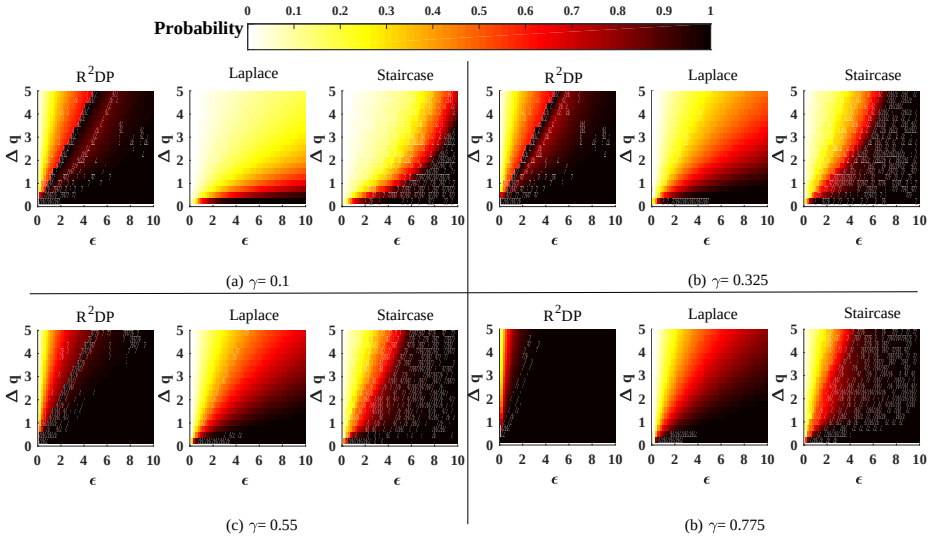


Figure 34: The R^{2}DP mechanism significantly outperforms the competing Laplace and the staircase mechanisms in maximizing the usefulness metric (an example of a utility metric with no known optimal PDF).

4.5.5 Numerical Analysis

We also demonstrate the effectiveness of R^{2}DP through numerical results based on Algorithm 2 (the ensemble R^{2}DP algorithm). In particular, Figure 34 depicts the corresponding usefulness (the probability of the results to be within a pre-specified error bound) of the R^{2}DP, the Laplace and the Staircase mechanisms. Figure 34 clearly demonstrates the fact that the R^{2}DP mechanism can significantly improve both already considered to be competing mechanisms. In particular, we observe the power of the R^{2}DP mechanism in generating very high utility results, e.g., results with more than 0.8 probability fallen inside only $\gamma = 0.1$ error-bound, owing to automatically searching a large search space of PDFs.

4.5.6 R^{2}DP and Other DP Mechanisms

In this section we briefly discuss the application of the R^{2}DP framework in two other well-known baseline DP mechanisms.

4.5.7 R^{2}DP Exponential Mechanism

The exponential mechanism was designed for situations in which we wish to choose the "best" response but adding noise directly to the computed quantity can completely destroy its value, such as setting a price in an auction, where the goal is to maximize revenue, and adding a small amount of positive noise to the optimal price (in order to protect the privacy of a bid) could dramatically reduce the resulting revenue [58]. The exponential mechanism is the natural building block for answering queries with arbitrary utilities (and arbitrary non-numeric range), while preserving differential privacy. Given some arbitrary range $\mathcal{R}$, the exponential mechanism is defined with respect to some utility function $u : \mathbb{N}^{|\mathcal{X}|} \times \mathcal{R} \to \mathbb{R}$, which maps database/output pairs to utility scores. Intuitively, for a fixed database x, the user prefers that the mechanism outputs some element of $\mathcal{R}$ with the maximum possible utility score. Note that when we talk about the sensitivity of the utility score $u : \mathbb{N}^{|\mathcal{X}|} \times \mathcal{R} \to \mathbb{R}$, we care only about the sensitivity of u with respect to its database argument; it can be arbitrarily sensitive in its range argument:

$$\Delta u \equiv \max_{r \in \mathcal{R}} \max_{x,y:\|x-y\|\leq 1} |u(x,r) - u(y,r)|.$$

The intuition behind the exponential mechanism is to output each possible $r \in \mathcal{R}$ with probability proportional to $\exp(\epsilon u(x,r)/\Delta u)$ and so the privacy loss is approximately:

$$\ln\left(\frac{exp(\epsilon u(x,r)/\Delta u)}{exp(\epsilon u(y,r)/\Delta u)}\right) = \epsilon[u(x,r) - u(y,r)/\Delta u] \leq \epsilon \tag{19}$$

The exponential mechanism is a canonical ϵ-DP mechanism, meaning that it describes a class of mechanisms that includes all possible differentially private mechanisms. However, the exponential mechanism can define a complex distribution over a large arbitrary domain, and so it may not be possible to implement the exponential mechanism efficiently when the range of u is super-polynomially large in the natural parameters of the problem [58]. This is the main restrictive aspect

of the exponential mechanism against leveraging different accuracy metrics. However, the exponential mechanism can benefit from the additional randomization of privacy budget, to handle the complexity (excessive sharpness) of the defined probability distribution. In particular, as we mentioned earlier, compound (or mixture) distributions arise naturally where a statistical population contains two or more sub-population which is the case for the exponential mechanism. Thus, we motivate the application of the R^{2}DP framework in designing exponential mechanisms with rather smooth but accurate distributions around each element in the range of u. However, further discussion on R^{2}DP exponential mechanism requires formal analysis, e.g., deriving the DP guarantee of such a mechanism.

4.5.8 R^{2}DP and Differential Privacy Relaxations

R^{2}DP can also be studied under various relaxations of differential privacy, e.g., (ϵ, δ)-differential privacy or Rényi Differential Privacy [129] which is a privacy notion based on the Rényi divergence [163]. These relaxations allow suppressing the long tails of the mechanism's distribution where pure ϵ-differential privacy guarantees may not hold. Instead, they offer asymptotically smaller cumulative loss under composition and allow greater flexibility in the selection of privacy preserving mechanisms [129]. In the following, we briefly discuss the application of R^{2}DP in two of such relaxed notions of the differential privacy .

4.5.8.1 R^{2}DP Gaussian Mechanism

A relaxation of ϵ-differential privacy allows an additional bound δ in its defining inequality:

Definition 4.5.1 ((ϵ, δ)-differential privacy [53])**.** *A randomized mechanism* $M : \mathsf{D} \times \Omega \to \mathsf{R}$ *is* (ϵ, δ)*-differentially private if for all adjacent* $d, d' \in \mathsf{D}$*, we have*

$$\mathbb{P}(M(d) \in S) \leq e^{\epsilon}\mathbb{P}(M(d') \in S) + \delta, \quad \forall S \subset \mathsf{R}. \tag{20}$$

This definition quantifies the allowed deviation (δ) for the output distribution of a ϵ-differentially

private mechanism, when a single individual is added or removed from a dataset. A differentially private mechanism proposed in [53] modifies an answer to a numerical query by adding the independent and identically distributed zero-mean Gaussian noise.

Given the definition of the $\mathcal{Q}$-function $\mathcal{Q}(x) := \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{u^2}{2}} du$, we have the following theorem [53, 105].

Theorem 4.5.11. *Let $q : \mathsf{D} \to \mathbb{R}$ be a query and $\epsilon > 0$. Then the Laplace mechanism $\mathcal{M}_q : \mathsf{D} \times \Omega \to \mathbb{R}$ defined by $\mathcal{M}_q(d) = q(d) + w$, with $w \sim \mathcal{N}(0, \sigma^2)$, where $\sigma \geq \frac{\Delta q}{2\epsilon}(K + \sqrt{K^2 + 2\epsilon})$ and $K = \mathcal{Q}^{-1}(\delta)$, satisfies (ϵ, δ)-DP.*

We define $\kappa_{\delta,\epsilon} = \frac{1}{2\epsilon}(K + \sqrt{K^2 + 2\epsilon})$, then the standard deviation σ in Theorem 4.5.11 can be written as $\sigma(\delta, \epsilon) = \kappa_{\delta,\epsilon} \Delta q$. It can be shown that $\kappa_{\delta,\epsilon}$ behaves roughly as $O(\ln(1/\delta))^{1/2}/\epsilon$. For example, to ensure (ϵ, δ)-differential privacy with $\epsilon = \ln(2)$ and $\delta = 0.05$, the standard deviation of the injected Gaussian noise should be about 2.65 times the ℓ_1-sensitivity of q.

Theorem 4.5.12. *The Gaussian Mechanism in Theorem 4.5.11 is $(\gamma, 2 \cdot \mathcal{Q}\big(\frac{\gamma}{\sigma(\delta,\epsilon)}\big))$-useful.*

Similar to our R^{2}DP Laplace mechanism, we can formulate an optimization problem for the R^{2}DP model using Gaussian mechanism. Therefore, using Theorems 4.5.11 and 4.5.12, we have the following.

Corollary 4.5.1. *Denote by u, the set of parameters for a probability distribution f_σ. Then, the optimal usefulness of an R^{2}DP Gaussian mechanism utilizing f_σ, at each quadruplet $(\epsilon, \delta, \Delta q, \gamma)$ is*

$$
\begin{aligned}
U_f(\epsilon, \delta, \Delta q, \gamma) = \max_{u \in \mathbb{R}^{|u|}} \big(1 - 2 \cdot \mathbb{E}_\sigma(Q\big(\tfrac{\gamma}{\sigma(\delta,\epsilon)}\big))\big) \\
\textit{subject to} \\
\max_{\forall S \in \mathsf{R}} \left\{ \tfrac{\mathbb{P}(\mathcal{M}_q(d,\sigma) \in S)}{\mathbb{P}(\mathcal{M}_q(d',\sigma) \in S)} \right\} = \epsilon, \\
\mathbb{E}_\sigma(Q(\epsilon\sigma - \tfrac{1}{2\sigma})) = \delta
\end{aligned}
\tag{21}
$$

4.5.8.2 R^{2}DP and Rényi Differential Privacy

Despite its notable advantages in numerous applications, the definition of (ϵ, δ)-differential privacy has the following two limitations.

First, (ϵ, δ)-differential privacy was applied to the analysis of the Gaussian mechanism [53]. In contrast to the Laplace mechanism (whose privacy guarantee is characterized tightly and accurately by ϵ-differential privacy), a single Gaussian mechanism satisfies a curve of $(\epsilon(\delta), \delta)$-differential privacy definitions [53]. Picking any one point on this curve may leave out important information about the mechanism's actual behavior [129].

Second, (ϵ, δ)-differential privacy also has limitations on the composition of differential privacy [123]. By relaxing the guarantee to (ϵ, δ)-differential privacy, advanced composition allows tighter analyses for compositions of (pure) differentially private mechanisms. Iterating this process, however, quickly leads to a combinatorial explosion of parameters, as each application of an advanced composition theorem leads to a wide selection of possibilities for $(\epsilon(\delta), \delta)$-differentially private guarantees.

To address these shortcomings, Rényi differential privacy was proposed as a natural relaxation of differential privacy in [129].

Definition 4.5.2 ((α, ϵ)-RDP)**.** *A randomized mechanism $M : \mathsf{D} \times \Omega \to \mathsf{R}$ is said to have ϵ-Rényi differential privacy of order α, or (α, ϵ)-RDP for short, if for if for all adjacent $d, d' \in \mathsf{D}$, we have $D_\alpha(M(d)||M(d')) \leq \epsilon$, where $D_\alpha(\cdot)$ is the (parameterized) Rényi divergence [163].*

Compared to (ϵ, δ)-differential privacy, Rényi differential privacy is a strictly stronger privacy definition. It offers an operationally convenient and quantitatively accurate way of tracking cumulative privacy loss throughout execution of a standalone differentially private mechanism and across many such mechanisms [129]. Next, we give the Rényi differential privacy guarantee of our R^{2}DP mechanism and show that the privacy loss of R^{2}DP under Rényi DP can significantly (asymptotically for small α) outperform Laplace, Gaussian and Random Response mechanisms.

Theorem 4.5.13. *If real-valued query q has sensitivity 1, then the R^{2}DP mechanism $\mathcal{M}_q$, leveraging MGF M, satisfies*

$$\begin{cases} (\alpha, \frac{1}{\alpha-1}\log\left[\frac{\alpha M(\alpha-1)+(\alpha-1)M(-\alpha)}{2\alpha-1}\right])\text{-}\textit{RDP}. & if\ \alpha > 1 \\ (1, M'(0)+M(-1)-1)\text{-}\textit{RDP}. & if\ \alpha = 1 \end{cases}$$

Proof. The above RDP guarantee follows Corollary 2 in [129] on the RDP guarantee of the classic Laplace mechanism. In particular, the above equations are derived using the following substitutions $exp(t/b) \rightarrow M(t)$ and $1/b \rightarrow M'(0)$ due to the second-fold randomization of b. □

4.5.9 Other Applications of R^{2}DP

R^{2}DP represents a very general concept which could potentially be applied in a broader range of contexts. In general, applying R^{2}DP to design more application-aware mechanisms may further improve the utility of many existing solutions [139]. We now briefly discuss some of the potential applications as follows.

R^{2}DP and Query-Workload Answering [111]. Given a workload (aka. a batch of queries), the matrix mechanism generates a different set of queries, called *strategy queries*, which are answered using a standard Laplace or Gaussian mechanism. The noisy answers to the workload queries can then be derived from the noisy answers to the strategy queries [110]. This two-stage process can result in a correlated noise distribution that preserves differential privacy and also increases utility.

Given a triplet (ϵ, query, metric), R^{2}DP can be applied to replace the Laplace or Gaussian mechanism for answering the strategy queries of the matrix mechanism. As a result, R^{2}DP will provide additional improvement in utility (in terms of the TotalError as defined in [110]) over the improvement already provided by the matrix mechanism. More specifically, we compare the total errors of Laplace and R^{2}DP mechanisms in Table 8 for specific workloads of interest (similar to those considered in [110]). These two workloads were analyzed in [110] using two n-sized query strategies, each of which can be envisioned as a recursive partitioning of the domain based on

the Haar wavelet [169]. We denote by $f(\epsilon, \Delta q)$ the improvement in the TotalError for applying an R^{2}DP noise instead of a Laplace noise in the matrix mechanism. For instance, leveraging the results of R^{2}DP (w.r.t. ℓ_1 or ℓ_2) shown in Section 4.4.2.2, for a workload of size $n = 6$, at $\epsilon = 2.3$, the improvement for range queries ($\Delta q = 36$) and predicate queries ($\Delta q = 64$) are $\sim$20% and $\sim$10%, respectively.

Table 8: Total error of matrix mechanisms comparison (with R^{2}DP vs. Laplace) – two workloads and two query strategies

TotalError		Matrix Strategies	
Mechanisms	Workload Queries	Binary Hierarchy of Sums	Matrix of the Haar Wavelet
Laplace	Range Queries	$\Theta(n^2 \log^3(n)/\epsilon^2)$	$\Theta(n^2 \log^3(n)/\epsilon^2)$
	Predicate Queries	$\Theta(n2^n \log^2(n)/\epsilon^2)$	$\Theta(n2^n \log^2(n)/\epsilon^2)$
R^{2}DP	Range Queries	$\Theta(f(\epsilon, n^2)n^2 \log^3(n)/\epsilon^2)$	$\Theta(f(\epsilon, n^2)n^2\epsilon^2 \log^3(n))$
	Predicate Queries	$\Theta(f(\epsilon, 2^n)n2^n \log^2(n)/\epsilon^2)$	$\Theta(f(\epsilon, 2^n)n2^n \log^2(n)/\epsilon^2)$

R^{2}DP and Composition. R^{2}DP may be applied for reducing the privacy leakage due to sequential or parallel querying over a dataset, of which the objective will be to maximize the number of compositions under a specified ϵ-differential privacy constraint.

R^{2}DP and Local Differential Privacy. In this context, R^{2}DP can be regarded as a new randomized response model. In particular, the randomized response scheme presented in [167] can be produced using R^{2}DP for the Bernoulli distribution when $b_0 \to 0$ and $b_1 \to \infty$. Therefore, designing more efficient local differential privacy schemes using R^{2}DP is an interesting future direction.

R^{2}DP for Continual Observation Applications. Providing differential privacy guarantees on data streams represents another important future direction for R^{2}DP. As an example, the multi-input multi-output (MIMO) systems process streams of signals originated from many sensors capturing privacy-sensitive events about individuals, and statistics of interest need to be continuously published in real time [56, 105], e.g., privacy-preserving traffic monitoring over multi-lane roads [27]. In this context, R^{2}DP can leverage the constraint related to the number of inputs and the number of outputs (e.g., the sensitivity of the output of MIMO filter G with m inputs and p outputs is proportional to the $\mathcal{H}_2$ norm of G which itself is an increasing function of m and p [140]) into its model to build more efficient differentially private mechanisms for the MIMO scenarios.

4.6 Summary

This work has proposed the R^2DP framework as a universal solution for optimizing a variety of utility metrics requested in different applications. It can automatically identify a distribution that yields near-optimal utility, and hence is more practical for emerging applications. Specifically, we have shown that a differentially private mechanism could be defined based on a random variable which is itself distributed according to some parameterized distributions. We have also shown that such a mechanism could explicitly take into account both the privacy requirements and the utility requirements specified by the data owner and data recipient, respectively. We have formally analyzed the privacy guarantee of R^2DP based on the well-known Laplace mechanism and formally proved the improvement of utility over the baseline Laplace mechanism. Furthermore, we discuss the potential of applying R^2DP to advanced algorithms. Finally, our experimental results based on six different utility metrics for statistical queries, machine learning and social network, as well as one privacy metric, have demonstrated that R^2DP could significantly improve the utility of differentially private solutions for a wide range of applications.

Chapter 5

*DPOD: D*ifferentially *P*rivate *O*utsourcing of Anomaly *D*etection with Optimal Sensitivity Learning

5.1 Introduction

There have been several studies on privately conducting anomaly detection over one or several datasets (vertically or horizontally partitioned datasets) in a centralized setting. However, to the best of our knowledge, here are very limited solutions to privately outsource the anomaly detection tasks to third-party managed security service providers (MSSP). MSSPs offer production cost advantages, updated technology and better trained expertise by specializing in the area and serving diverse range of clients. Despite all the benefits of outsourcing, organizations are still reluctant to share their data with third parties mainly due to privacy concerns over the sensitive information contained in such data. For example, important network configuration information, such as potential bottlenecks of the network, may be inferred from network traces and subsequently exploited by adversaries to increase the impact of a denial of service attack [149].

Recently, differential privacy has been widely recognized as the state-of-the-art [52, 55] privacy notion which provides protection by requiring the presence of any individual's data in the input to only marginally affect the distribution over the output, provides strong protection against adversaries with arbitrary background knowledge about the individuals. Popular approaches to differential privacy, such as the Laplace and exponential mechanisms, calibrate randomised smoothing through global sensitivity of the target non-private function. Bounding such sensitivity is often a prohibitively complex analytic calculation. Unfortunately, two prohibitive issues stem from applying DP to anomaly detection tasks. First, in many applications, from collaborative filtering [126], and Bayesian inference [45] to anomaly detection [124], the principal challenge in ensuring differential privacy is to bound the sensitivity. For instance, Mcsherry et al. [124] proposed a differentially private anomaly detection analysis for network traces by benchmarking with the Privacy Integrated Queries (PINQ) [123]. Unfortunately, applying PINQ directly to the network traces through the queries pertains a serious privacy violation. Specifically, one individual may contribute several records to a network trace resulting in a sensitivity value larger than one, whereas the privacy integrated queries (PINQ) assumes a sensitivity equal to one (packet level protection which obviously sacrifices the privacy of customers who contribute more than one packet into a network trace).

Second, anomaly detection aims to identify instances that are apparently distant from other instances, and the objective of differential privacy is to conceal the presence (or absence) of any particular instance, anomaly detection and privacy protection are therefore intrinsically conflicting tasks. Therefore, differential privacy is inherently incapable of both accurately and privately identifying anomalies (Hafiz et al. [6] formalized the strict trade-off between the differential privacy constraint and the error in computing anomalous records). This conflict is the main obstacle against a generic solution. Therefore, each of the existing works tackles the challenges entailed with their assumed/relaxed model of the problem. In particular, [17, 19, 141, 6, 7] propose methods for searching anomalous records in a rather restricted setting (centralized setting where the data analyst can access to the raw data and the recipient of the detection is untrusted). In particular, as

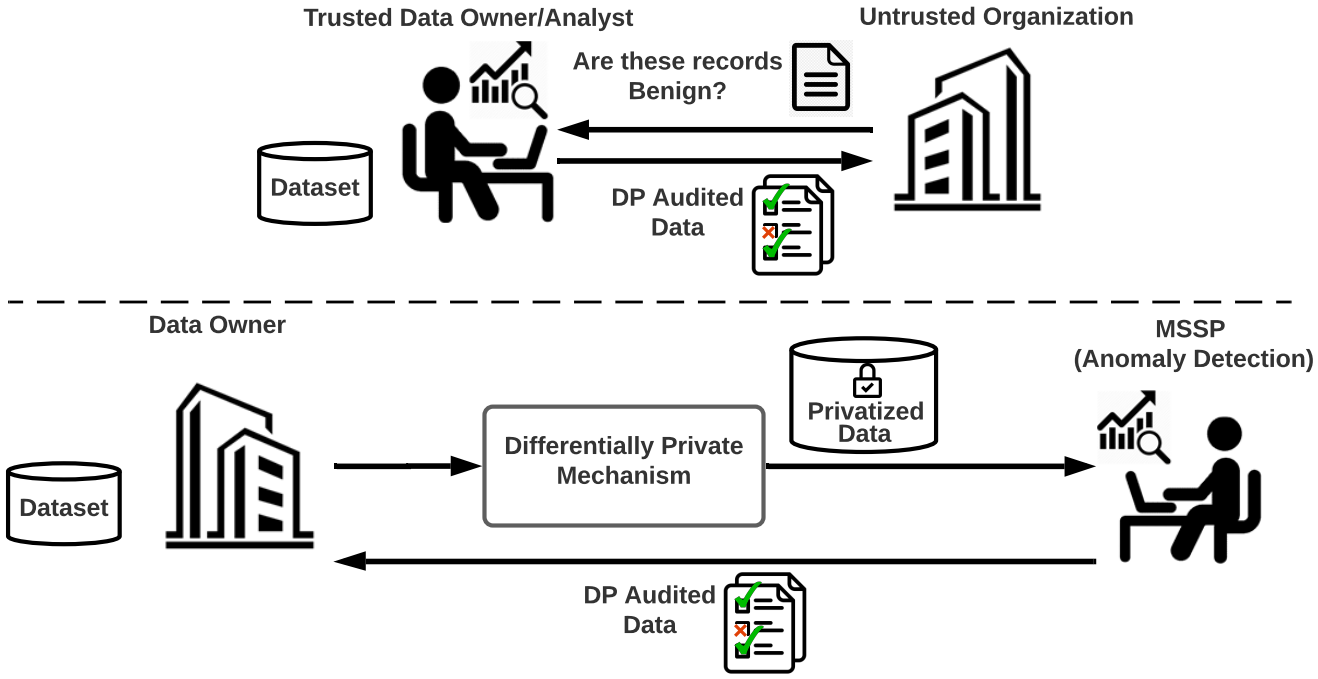


Figure 35: System Model of existing works [17, 19, 141, 6, 7] (top), and of DPOD (down)

depicted in Figure 35, the system model of such works assumes a trusted data owner/analyst who has access to the database, and it answers the anomaly identification queries using a privacy preserving mechanism. In fact, in these settings the data owner and the analyst are the same entities and the goal of the privacy mechanism is to protect the privacy of analysis results against, e.g., a manufacturer of IoT devices.

5.1.1 DPOD: A Novel Framework

To address the two aforementioned issues, we propose a framework, namely, *D*ifferentially *P*rivate *O*tsourcing of *A*nomaly *D*etection (DPOD), which borrows an appropriate notion of privacy with the following two important properties (CCS'19 [6]). First, the more outlying (or non-outlying) a record is, the higher the accuracy the privacy mechanism can achieve for anomaly identification. Second, all the benign records should have DP like privacy guarantee for the same value of privacy parameter. This notion of privacy is illustrated in Figure 36.

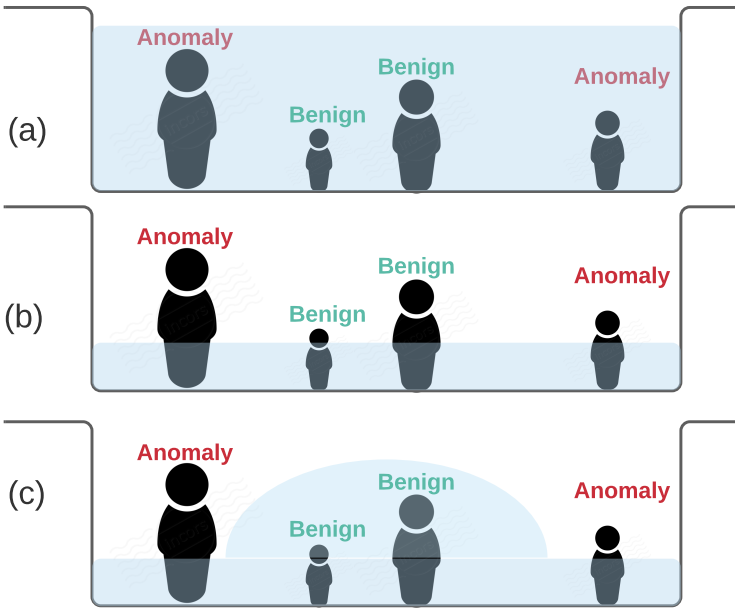


Figure 36: The water height probabilistically represents the amount of the noise injected to the data (the heights) of each individual when the outsourcing scheme acquirers (a) a reasonably high sensitivity, (b) a weak sensitivity, and (c) the DPOD's sensitivity

Specifically, we borrow a naturally-relaxed definition of differential privacy, namely the *Random Differential Privacy* [84, 152], which leverages a sampler for estimating sensitivity of the non-private mechanisms. Specifically, we construct the probability distribution of the dataset using the computed anomaly scores (which is a valid representation of the PDF). Next, the sensitivity sampler (based on the constructed PDF) is applied. Rubinstein et al. [152] have shown that for any mechanism that is ϵ-differentially private under bounded global sensitivity, such a mechanism automatically achieves (ϵ, γ)-random differential privacy [84], without any target-specific calculations required. Therefore, we can significantly reduce the required distortion in providing a strong level of protection to records that are with high probability benign (results in weaker protection to those that are most likely malicious).

However, adopting this notion in outsourcing setting is not straightforward because the data owner and the data analyst (MSSP) are two separated entities, each of them cannot accomplish his/her task without the other one's output. Therefore, a data owner through DPOD iteratively and

efficiently interacts with the MSSP to construct the PDF of the sensitivity sampler. Our solution for a data owner starts with the naïve solution (uniform distribution; or equal privacy for all records), and probabilistically refines and updates the PDF over time using the returned anomaly scores (see Section 5.3 for the details).

5.1.2 Contributions

Specifically, we make the following contributions:

1. DPOD provides the first practical differentially private anomaly detection in outsourcing setting which includes both single (one data owner) and inter-domain(multiple data owners) use cases. The problem of practicality has been raised in several existing works, e.g., the inherent conflict between anomaly detection and DP.

2. We propose a novel differential privacy mechanism for DPOD, including (1) leveraging a suitable relaxed-notion of DP, called *Random Differential Privacy (RDP)*, which perfectly satisfies the two essential requirements of a suitable privacy notion for DP anomaly detection to address the inherent conflict discussed, (2) proposing a formal methodology based on the methodology of Rubinstein et al. [152] (called Pain-free) which leverages a sensitivity sampler over the uniform distribution for all possible data entries, (3) further boosting the accuracy by updating the true distribution of the data over time using the computed anomaly scores, and leveraging this distribution into the sensitivity sampler.

3. We formally benchmark DPOD under the Laplace mechanism for network, IoT and credit card anomaly (fraudulent) detection.

4. Our experimental results demonstrate that DPOD significantly improves the accuracy of the anomaly detection compared to the baseline Laplace distribution (as the most polpular but not efficient choice), and the pain-free solution [152]).

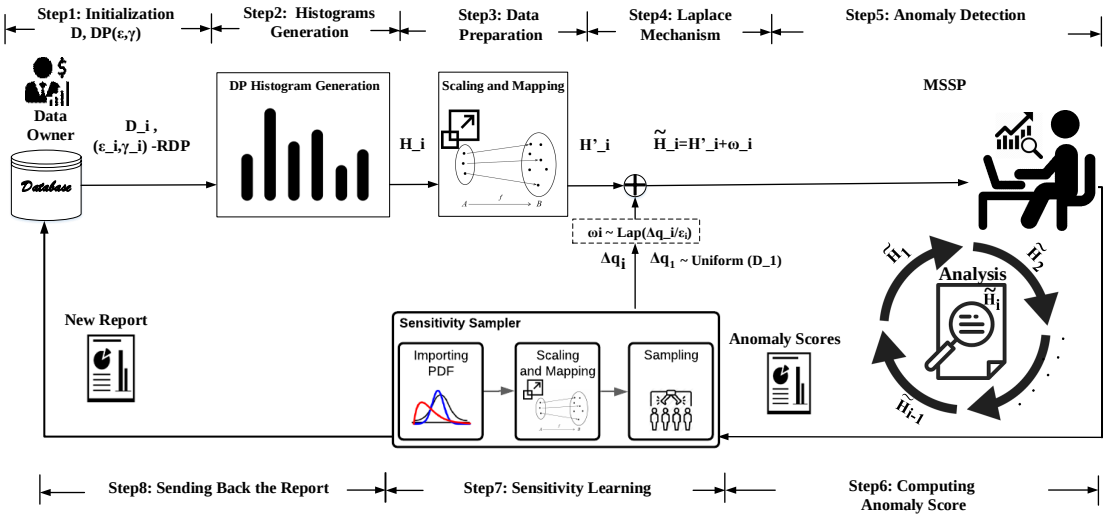


Figure 37: The high level overview of the DPOD framework

The rest of the chapter is organized as follows. Section 5.2 provides some related background. Section 5.3 defines the DPOD framework. Section 5.4 presents the experiments and Section 5.5 concludes the chapter.

5.2 Overview

5.2.1 The System Model

We consider a data owner who is interested in sharing with an MSSP his/her data over time to monitor anomalous activities, and is interested in applying a privacy preserving mechanism with the guarantee of random differential privacy. Specifically, privatized histogram versions $\tilde{H}_1, \tilde{H}_2, \cdots, \tilde{H}_t$ from data excerpts $D_1, D_2, \cdots, D_t$, are shared with the MSSP to conduct t analyses. In this context, we make the following assumptions (similar to those found in most existing works [134, 98, 49]).

i) The MSSP is a honest-but-curious analyst (in the sense that the MSSP generates anomaly scores trustworthy) who can observe $\tilde{H}_1, \tilde{H}_2, \cdots, \tilde{H}_t$.

i) The goal of the adversary is to figure out if a given individual's data exist in $\tilde{H}_1, \tilde{H}_2, \cdots, \tilde{H}_t$.

i) Finally, we assume the communication between the data owner and the analyst is over a secure channel, and we do not consider integrity or availability issues (e.g., a malicious adversary may potentially alter or delete the analysis report).

5.3 The DPOD Framework

In this section, we present the DPOD framework and its main building blocks. Our approach is depicted in Figure 37 and detailed below.

5.3.1 The DPOD Approach

The detailed workflow for DPOD is as following.

Step 1: The data owner specifies the random differential privacy parameters ϵ, γ and inputs his/her data D_i to the histogram function.

Step 2: The histogram function transforms the multi-dimensional data D_i to a histogram version H_i according to the requirements of the MSSP (determining the bins boundaries).

Step 3: The generated histogram data is now fed into the data preparation function which includes an scaling and a mapping function. These two functions sort the values in H_i based on their anomaly scores and resulting in H'_i. We note the first iteration of DPOD is slightly different from other iterations since the anomaly scores are not yet computed, and this step is not applied for the first iteration.

Step 4: The computed Laplace noise $\omega(\Delta q_i/\epsilon_i)$ is added to H'_i to generate the DP version $\tilde{H'}_i$. The sensitivity value of this noise in the first iteration acquires the Pain-Free algorithm (sensitivity sampler using a uniform distribution), and for other iterations, Δq_i is given by the sensitivity sampler output (Step 7).

Step 5: The MSSP analyzes $\tilde{H'}_i$ to identify the anomalous activities.

Step 6: The MSSP generates the anomaly scores.

Step 7: The set of anomaly scores is then fed into the Sensitivity Sampler block to calculate the sensitivity of the next round Δq_{i+1}. This block first applies the same functions as the data preparation, i.e., scaling and mapping, to the probability distribution function defined using the scores. These two functions are essential since they transform the various version of the data to a format with ascending anomaly scores. Therefore, the calculated sensitivity guarantees the privacy of benign records (which are now smaller after scaling and mapping), with a high probability, and reveals the anomalous records (which are now larger after scaling and mapping).

Step 8: The analysis report is sent back to the data owner.

Next, we define the privacy property for the multi-view solution.

Table 9: Notations symbols and their descriptions

Notations	Descriptions
$\epsilon_1, \epsilon_2, \cdots, \epsilon_n$	Privacy constraints from data owners
Δq_i	Updated sensitivity at iteration i
b	Noise parameter in Laplace mechanism
ω	Noise
$\tilde{H}_i$	DP histogram at iteration i

5.3.2 Building Blocks

In this section, we introduce the main building blocks for our DPOD framework, namely, the *data preparation*, and the *sensitivity sampler*.

5.3.2.1 Data Preparation

The main objective of the DPOD approach is that by excluding the set of anomalous record from the set of privacy-sensitive data, privacy preservation of the benign records, and anomaly detection

in MSSP side will be simultaneously enabled. This can be done only if the anomalous records contribute relatively larger values to the dataset resulting in a larger sensitivity. Therefore, to bound the global sensitivity, our key idea is to map (according to the anomaly scores of the previous round) and scale (to make the mapped data similar to the original data values) each excerpt of the data.

5.3.2.2 Sensitivity Sampler

The main idea of Sensitivity Sampler is that for each extended-database observation of D P^n, we induce i.i.d. observations $G_1, \cdots, G_m \in \mathbb{R}$ of the random variable $G = \|f(\mathsf{D}_{1\cdots n}) - f(\mathsf{D}_{1\cdots n-1;n+1})\|$. While observing the sensitivity of the target mapping which is the histogram function $f : \mathsf{D} \rightarrow H$, we estimate w.h.p. the value of the sensitivity that can achieve random differential privacy. If we knew the full CDF of G (all the iterations after the first iteration), we would simply invert this CDF to determine the level of sensitivity for achieving any desired γ level of random differential privacy: higher confidence would invoke higher sensitivity and therefore lower utility. However as we cannot possess the true CDF in the first iteration, we resort to uniformly approximate it w.h.p. using the empirical CDF induced by the sample $G_1, \cdots, G_m$. The guarantee of uniform approximation is derived from the empirical process theory. Finally, Algorithm 3 summarizes the aforementioned steps.

5.4 Experiments

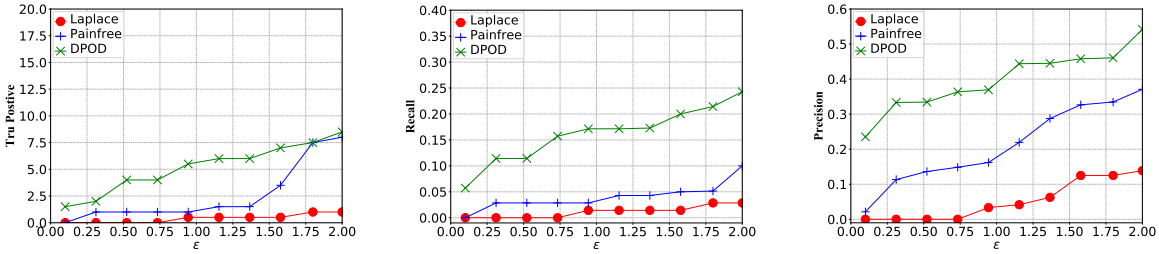
In this section, we experimentally evaluate the performance of the DPOD framework using six different well-known datasets, i.e., smart home event and sensor dataset, parking birmingham dataset, individual household electric power consumption dataset, breast cancer, credit card clients dataset, and KDD Cup 1999 dataset. The selected datasets cover various domains, such as, IoT, medical, smart grid, and finance, etc. Table 10 provides the characteristics of the datasets.

Input : Dataset D, Privacy budget ϵ
Output: Set of identified Anomalies which is ϵ-DP
$\Delta q_O \leftarrow$ Global Sensitivity (D)
i=0
while $\epsilon > 0$ **do**
    i=i+1
    $\Delta q_E \leftarrow$ Estimated Sensitivity (an exemplary solution to address this problem of sensitivity learning by using the approach in [26])
    $\epsilon_O \leftarrow$ Find optimal budget for outsourcing of next round using Kelly criterion
    $\epsilon = \epsilon - \epsilon_o$
    $\tilde{H}_i = \mathcal{M}(D, \epsilon_O, \Delta q_O)$: Appropriate DP mechanism $\mathcal{M}$ to release private data type required by data analyst using the input parameters, e.g., DP-Histogram mechanism
    $\mathcal{R} = \mathcal{A}(\tilde{H}_{1,2,\cdots,i})$: Analyzing all previous received datasets to optimally identify anomalous records
    $\Delta q_o \leftarrow$ Outsourcing learned sensitivity for next round using all previous rounds of analysis
end
return $\mathcal{R}$

Algorithm 3: DPOD Algorithm

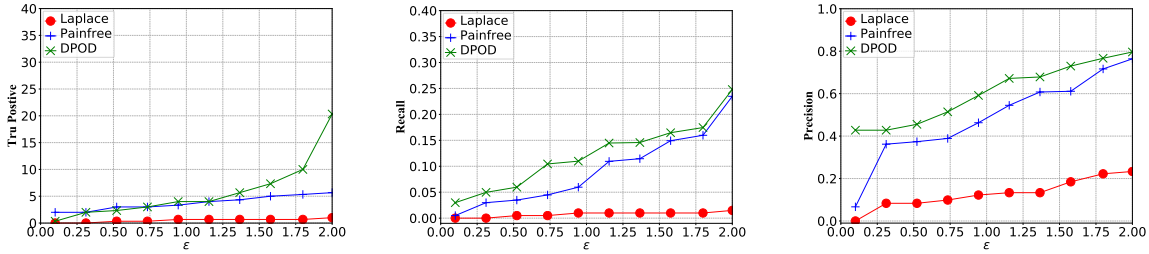
Table 10: Summary of the Datasets for DPOD evaluation

Databset	**Size (MB)**	**# of Attributes**
IoT	20000	12 events + 10 sensors
Parking	35,718	4
Electric consumption	2	9
Breast cancer	286	9
Credit card	30,000	23
KDD	494,021	42



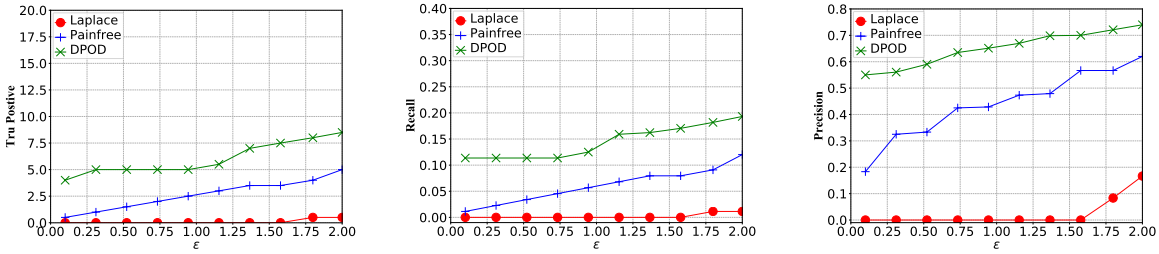
(a) ϵ v.s. True Positive (b) ϵ v.s. Recall (c) ϵ v.s. Precison

Figure 38: Evaluation of three mechanisms with parking dataset for different accuracy metrics



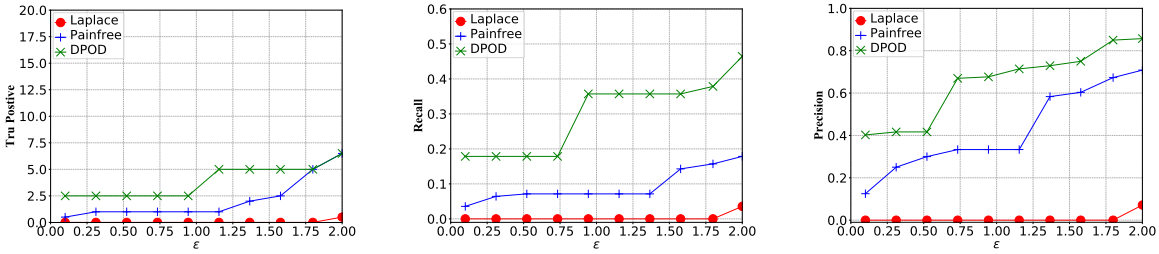
(a) ϵ v.s. True Positive (b) ϵ v.s. Recall (c) ϵ v.s. Precison

Figure 39: Evaluation of three mechanisms with electric consumption dataset for different accuracy metrics



(a) ϵ v.s. True Positive (b) ϵ v.s. Recall (c) ϵ v.s. Precison

Figure 40: Evaluation of three mechanisms with credit card dataset for different accuracy metrics

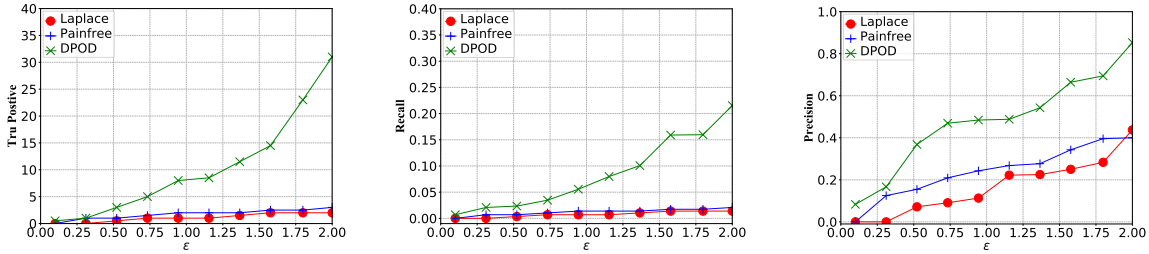


(a) ϵ v.s. True Positive (b) ϵ v.s. Recall (c) ϵ v.s. Precison

Figure 41: Evaluation of three mechanisms with KDD dataset for different accuracy metrics

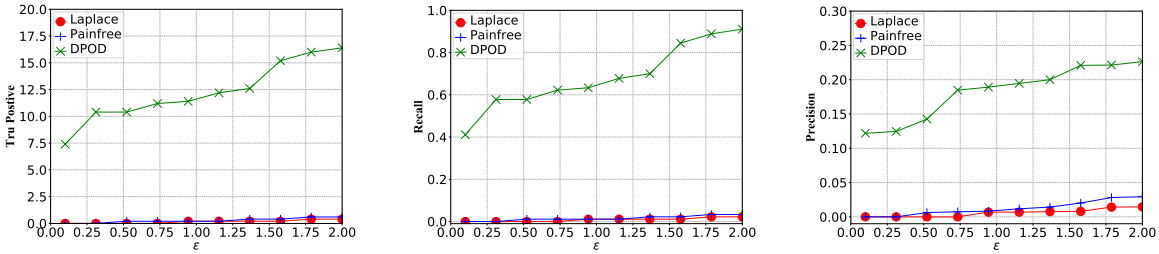
5.4.1 Experimental Setting

We perform all the experiments and comparisons on both privacy parameters and anomaly detection parameters. Our solutions, Painfree and DPOD, are compared with the baseline solution, Laplace mechanism. Our objective is to verify the following two properties about the performance of the



(a) ϵ v.s. True Positive (b) ϵ v.s. Recall (c) ϵ v.s. Precison

Figure 42: Evaluation of three mechanisms with the traffic data for different accuracy metrics



(a) ϵ v.s. True Positive (b) ϵ v.s. Recall (c) ϵ v.s. Precison

Figure 43: Evaluation of three mechanisms with Densities for different accuracy metrics

DPOD framework w.r.t. all six datasets: (1) DPOD preserves the privacy of the benign records, and (2) DPOD performs strictly better than well-known baseline mechanisms, e.g, Laplace and Pain-Free (also our proposal but a trivial one) mechanisms. I would like to thank Han Wang for her help in conducting the following results.

5.4.1.1 Privacy Parameters

In the first set of our experiments, we examine the benefits of DPOD on varying privacy parameters, i.e., ϵ and γ, against three well known accuracy metrics, true positive, recall, and precision. For all the experiments on privacy parameters, we choose anomaly detection parameters, i.e., the threshold and evaluation rounds, to be 0.7 and 5, respectively.

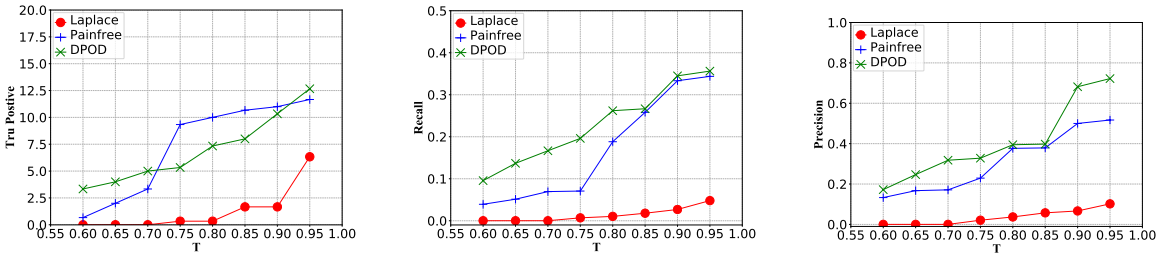
For each dataset, we first sample the sensitivity parameter, Δq from the uniform distribution

defined over all possible values of a dataset. For each iteration, we update the underlying distribution using the anomaly scores computed by an outlier analyst. In all the experiments, the outlier analyst performs anomaly detection based on Kolmogorov Smirnov (KS) test in the Matlab implementation [100]. We measure the error of a privacy preserving mechanism (which is a randomized algorithm) as its probability of outputting the wrong answer—recall that in the case of AIQ, there are only two possible answers, i.e. 0 or 1. For each AIQ for a fixed record, we estimate the error by the average number of mistakes over m trials. Thus, for our experiments we choose m to be 10000.

5.4.2 Anomaly Detection Parameters

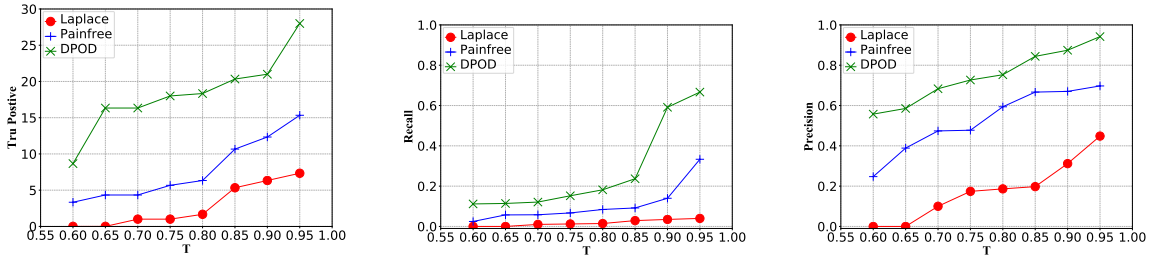
We compare how DPOD behaves under different detection thresholds and evaluation rounds from the outlier analyst. The threshold is the parameter that controls the generally controls using basic statistical functions, i.e., count and average. The dataset comes from a sensor network experiment carried out in the Mitsubishi Electric Research Laboratories (MERL) and described in [168]. MERL has collected the motion sensor data from a network of over 200 sensors for a year and the dataset contains over 30 million raw motion records. To illustrate the query performance with different sensitivities, we create the queries based on a subset of the data including aggregated events that are recorded by closely located sensors over 5-minute intervals. We follow this way to form 10 input signals corresponding to 10 spatial zones (each zone is covered by a group of sensors). Since each individual can activate several sensors and travel through different zones, we define moving average functions with arbitrary sensitivity values, e.g., $\Delta q \in [0.1, 5]$. For instance, we could be interested in the summation of the moving averages over the past 30 min for zones 1 to 4.

Summary. DPOD can generate better results than most of the well-known solutions. However, taking different factors in anomaly detection, e.g., data dependency (which determines the range of the sensitivity), scatterness and diversity, the improvement rate varies. In particular, we see that for credit fraud and parking datasets, DPOD for some of the anomalous records give smaller errors. We explain this deviation using the Credit Fraud dataset as an example. The aforementioned



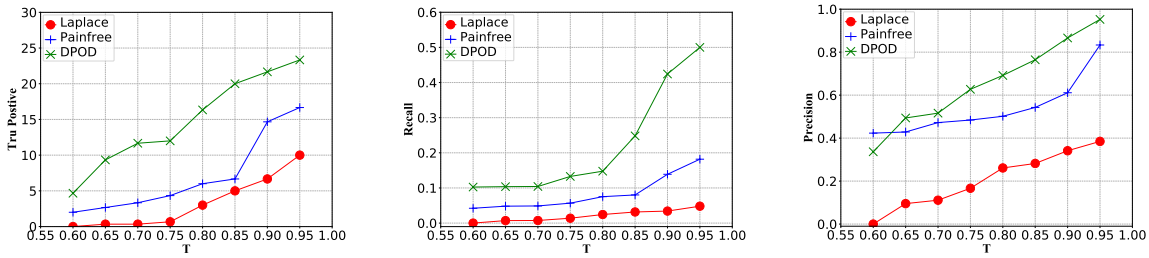
(a) Detection Threshold (b) Detection Threshold (c) Detection Threshold

Figure 44: Evaluation of three mechanisms with IoT data for different accuracy metrics



(a) Detection Threshold (b) Detection Threshold (c) Detection Threshold

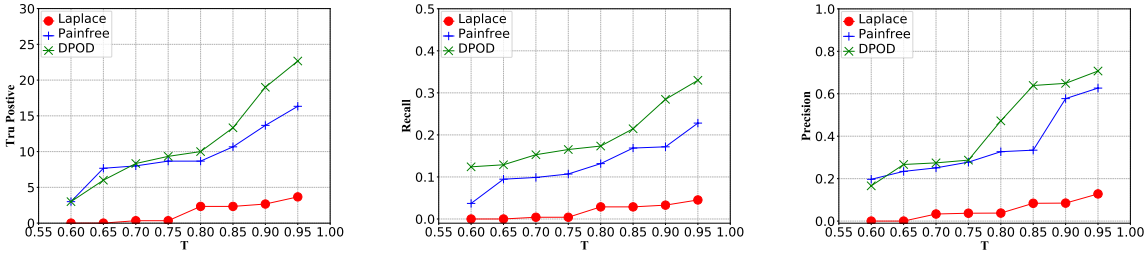
Figure 45: Evaluation of three mechanisms with parking data for different accuracy metrics



(a) Detection Threshold (b) Detection Threshold (c) Detection Threshold

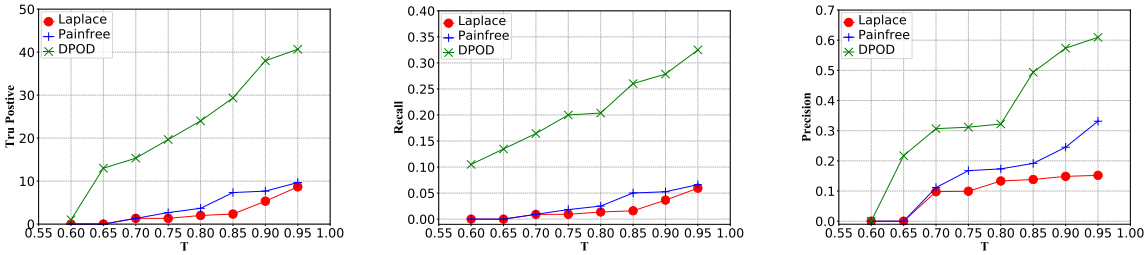
Figure 46: Evaluation of three mechanisms with electric consumption data for different accuracy metrics

deviation in the error occurs whenever the anomalous record is not unique, which is typically rare. The reason DP-mechanism's error remains constant in most cases is that the anomalies lie in a very sparse region of space and mostly do not have any duplicates (i.e., other records with the same value).



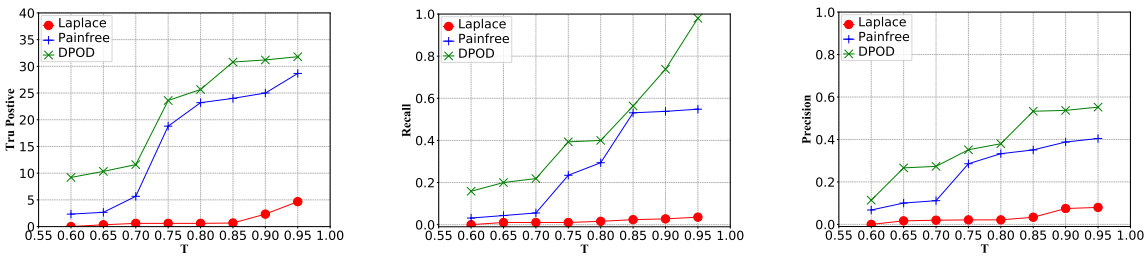
(a) Detection Threshold (b) Detection Threshold (c) Detection Threshold

Figure 47: Evaluation of three mechanisms with breast cancer data for different accuracy metrics



(a) Detection Threshold (b) Detection Threshold (c) Detection Threshold

Figure 48: Evaluation of three mechanisms with credit card data for different accuracy metrics

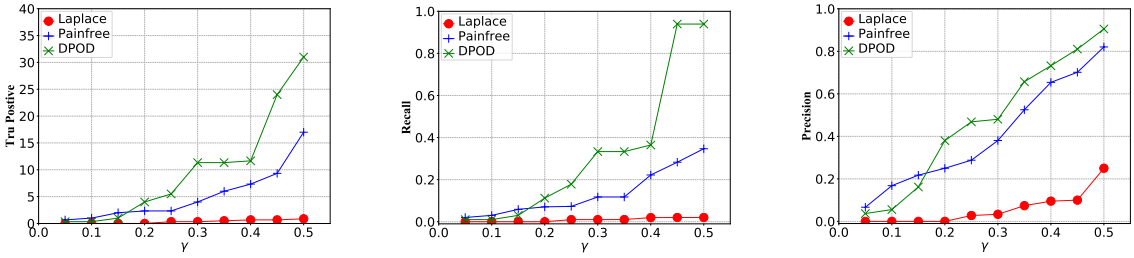


(a) Detection Threshold (b) Detection Threshold (c) Detection Threshold

Figure 49: Evaluation of three mechanisms with KDD data for different accuracy metrics

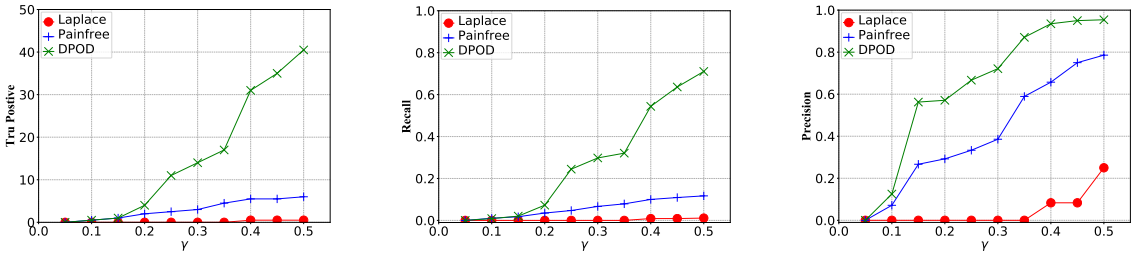
5.5 Summary

We propose a novel solution called DPOD to address the problem of privacy-preserving outsourcing of anomaly detection by decreasing the privacy of the anomalies through communication between the data owners and the data analysts. We show that DPOD can significantly improve the accuracy of the analysis for the data with abnormal behaviour while preserving the privacy of the data with



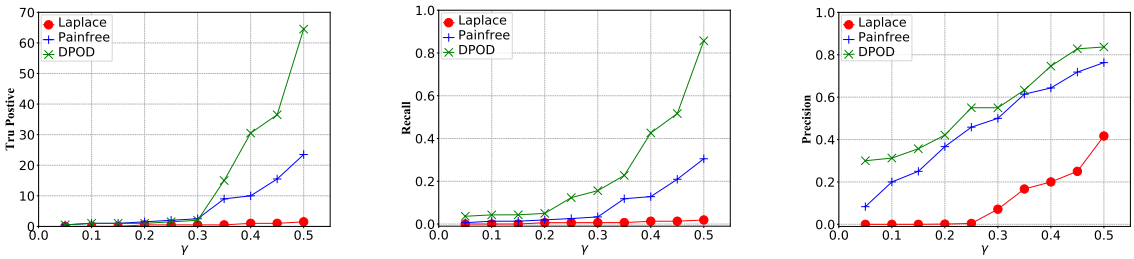
(a) Gamma (b) Gamma (c) Gamma

Figure 50: Evaluation of three mechanisms with IoT data for different accuracy metrics



(a) Gamma (b) Gamma (c) Gamma

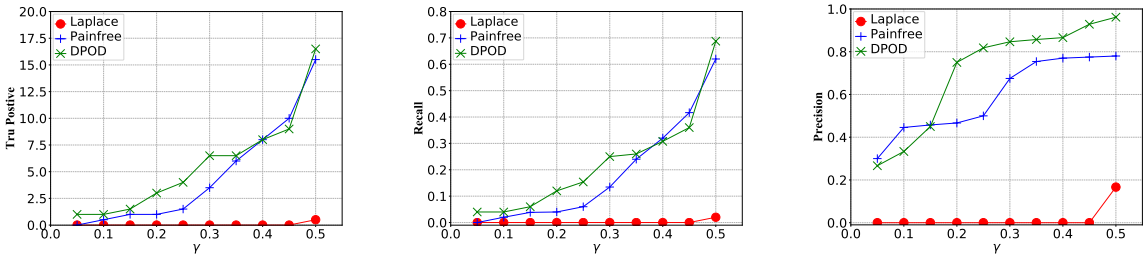
Figure 51: Evaluation of three mechanisms with parking data for different accuracy metrics



(a) Gamma (b) Gamma (c) Gamma

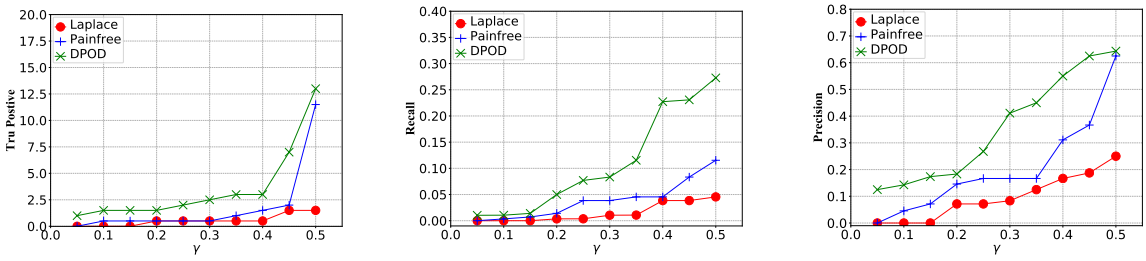
Figure 52: Evaluation of three mechanisms with electric consumption for different accuracy metrics

normal behavior. The system receives the input from the data owners and the data analysts. Each data owner spends a portion of his/her privacy budget to build a first estimation of the sensitivity. Each data analyst provides owners with an updated sensitivity value using the calculated anomaly scores of each record. The framework can be used in settings with one or more data owners.We formally benchmark DPOD under DP-Histogram publishing and showcase the improvements in



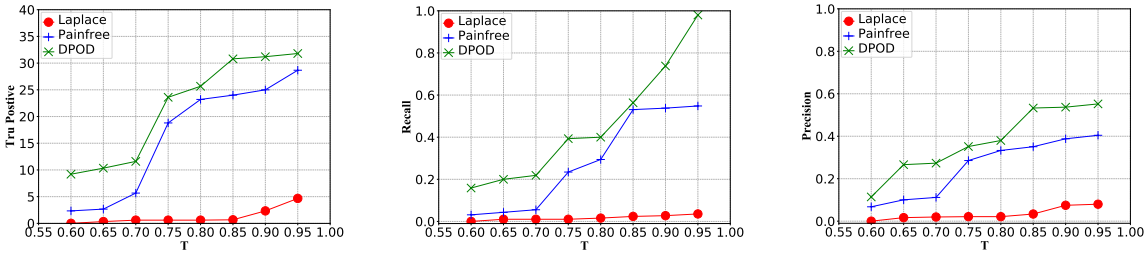
(a) Gamma (b) Gamma (c) Gamma

Figure 53: Evaluation of three mechanisms with breast cancer data for different accuracy metrics



(a) Gamma (b) Gamma (c) Gamma

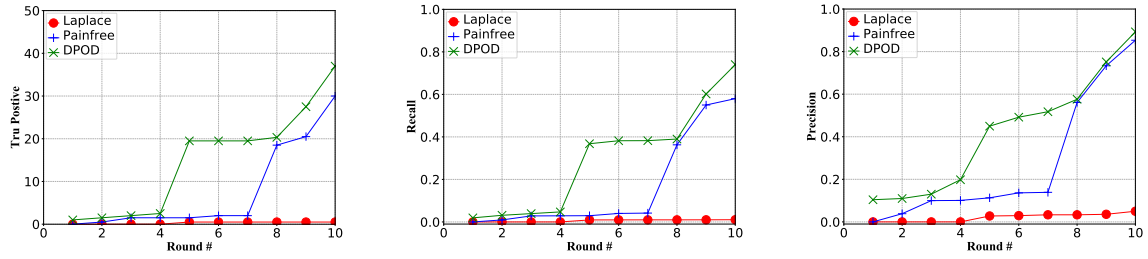
Figure 54: Evaluation of three mechanisms with KDD data for different accuracy metrics



(a) Gamma (b) Gamma (c) Gamma

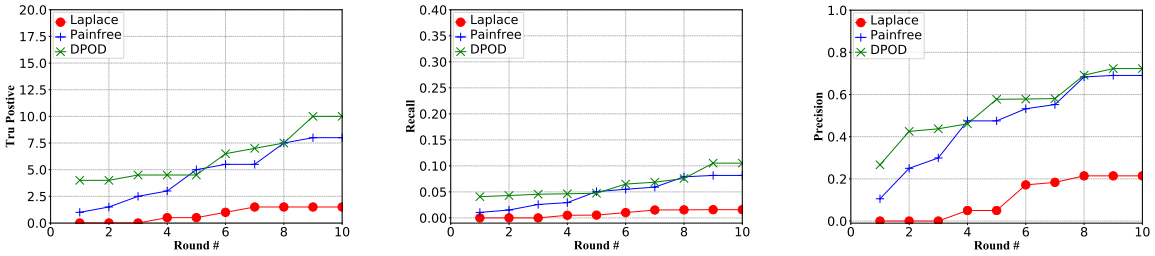
Figure 55: Evaluation of three mechanisms with Densities for different accuracy metrics

the performance.



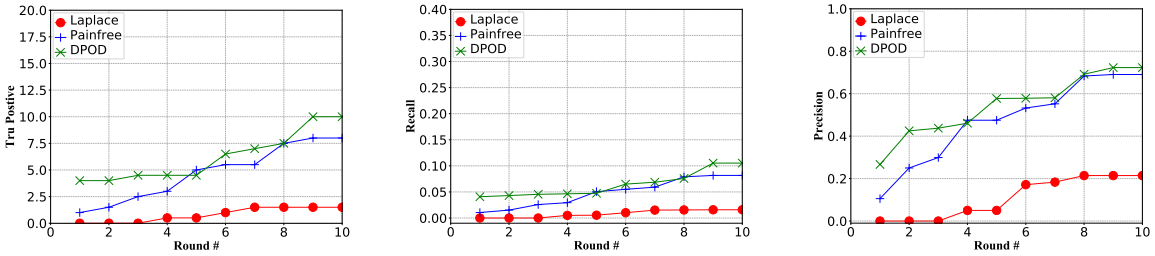
(a) Round (b) Round (c) Round

Figure 56: Evaluation of three mechanisms with IoT data for different accuracy metrics



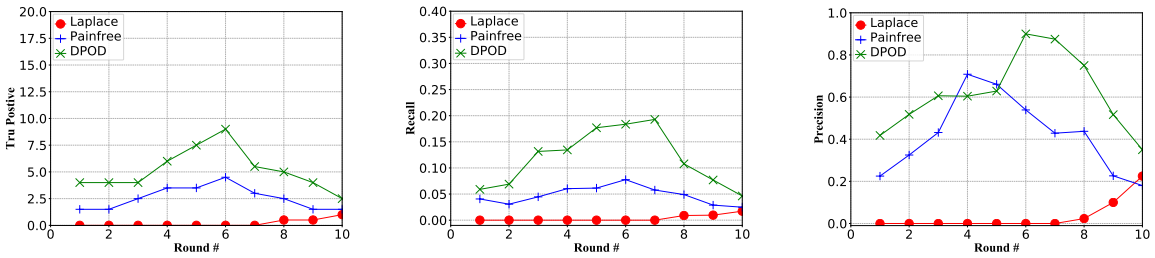
(a) Round (b) Round (c) Round

Figure 57: Evaluation of three mechanisms with parking data for different accuracy metrics



(a) Round (b) Round (c) Round

Figure 58: Evaluation of three mechanisms with electric consumption data for different accuracy metrics



(a) Round (b) Round (c) Round

Figure 59: Evaluation of three mechanisms with breast cancer data for different accuracy metrics

Chapter 6

Conclusion & Future Works

While pursuing better utility by discovering knowledge from the data, individual's privacy may be compromised during an analysis: corporate networks monitor their online behavior, advertising companies collect and share their private information, and cybercriminals cause financial damages through security breaches. In this topic, we investigate the possible three different frameworks in some well-known settings outlined in the following. The specific problems we explore in this dissertation include privacy preserving tool in both *local setting* (optimized application-aware mechanism design), and *outsourced setting* (network trace analysis and intrusion detection system). Specifically, in the first work of this dissertation, we have proposed a multi-view anonymization approach mitigating the semantic attacks on CryptoPAn while preserving the utility of the trace. The second work proposes the R^2DP framework as a universal solution for optimizing a variety of utility metrics requested in different applications. Finally, the third work we propose a novel framework called DPOD on privacy preserving Anomaly detection which has numerous applications in a very wide variety of domains such as data cleaning, fraud detection, financial markets, intrusion detection, and law enforcement.

Future works in this direction include building utility-maximized secure but ethical algorithms (privacy preserving, fair [2] and accountable) for a variety of generic tools including deep learning,

[2]The state-of-the-art AI learned model exhibits discrimination against some demographic group, perhaps based on race or gender.

cybersecurity and computational learning theory, and applications including health data monitoring and analysis, cloud computing and safe networking. Satisfying all the constraints pertained to these properties together, i.e., privacy, utility, fairness, etc, is shown to be contradicting in many cases and requires novel approaches to be proposed so that preserving one property minimally impacts the others. Fortunately, designing ethical algorithms is extremely relevant to several important areas of computer science and engineering such as deep learning, cloud computing and networking, big data, and online social networking. As shown in my research plan (Fig. 60), I plan to embrace this exciting opportunity for interdisciplinary research and collaboration.

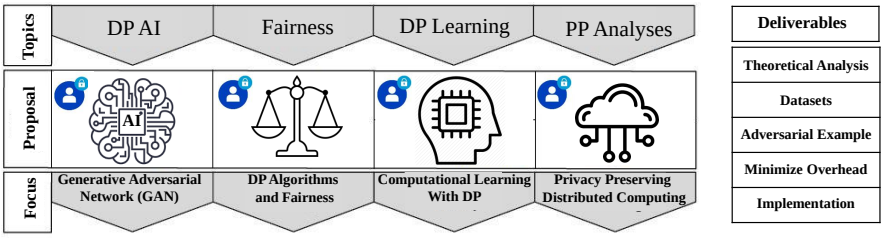


Figure 60: An Overview of the Research Agenda in "Ethical Algorithms".

6.1 AI with Differential Privacy and Fairness

Machine learning techniques based on neural networks are achieving remarkable results in a wide variety of domains. Often, the training of models requires large, representative datasets, which may be crowdsourced and contain sensitive information. The objective of this proposal in line with [ACM CCS' 16] is to build a differentially private neural network with minimal utility loss due to noise perturbation. Our key idea is to apply and adjust the R^2DP technique to achieve such optimality. Furthermore, I am very interested in analyzing generative adversarial networks (GAN) [NIPS' 14] [1] from privacy perspective (esp. Differential Privacy) and evaluate its privacy strength. In addition, I would like to consider other classes of deep networks that are not considered by Abadi et al. [ACM CCS' 16]. In particular, the naïve methodology of Abadi et al. [ACM CCS'

[1] Generative Adversarial Networks (GAN) are algorithmic architectures that use two neural networks, pitting one against the other (thus the "adversarial") in order to generate new, synthetic instances of data that can pass for real data.

16] demonstrates promising results for the MNIST [1] and the CIFAR-10 [2] datasets, but I see many opportunities for new research, for example in applying the R^2DP technique to LSTMs used for language modeling tasks. The accuracy improvement owing to applying the R^2DP technique is especially important (as also motivated in [ACM CCS' 16]) because many training datasets are much larger than those of MNIST and CIFAR10; and the accuracy should leverage the size of the dataset. Another interesting track of research is to investigate whether privacy and *fairness* [3] can be simultaneously achieved by a single classifier in several different models. Some of the earliest works on fairness in algorithm design defined fairness as a guarantee of similar outputs for "similar" input data, a notion with tight technical connections to differential privacy. Therefore, a very interesting research question is whether tensions exist between differential privacy and statistical notions of fairness, namely, Equality of False Positives and Equality of False Negatives (EFP/EFN).

6.2 Computational Learning Theory and Differential Privacy

Another viable track of research is to release synthetic databases that are useful for accurately answering large classes of ML queries while preserving differential privacy [J.ACM' 13]. Specifically, Blum et al. [J.ACM' 13] demonstrate that, ignoring computational constraints, it is possible to give such a mechanism. Unfortunately, the state-of-the-art is not yet able to release even simple classes of queries (such as intervals and their generalizations) over continuous domains with worst-case utility guarantees while preserving differential privacy. My proposal is to reduce computational overhead of such algorithms using GAN technology mentioned earlier. The objective of this proposal is to build a hybrid model of GAN [NIPS' 14] and the exponential mechanism of

[1] The MNIST database (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems.

[2] The CIFAR-10 dataset (Canadian Institute For Advanced Research) is a collection of images that are commonly used to train machine learning and computer vision algorithms. It is one of the most widely used datasets for machine learning research.

[3] In machine learning, a given algorithm is said to be fair, or to have fairness if its results are independent of some variables we consider to be sensitive and not related with it (f.e.: gender, ethnicity, sexual orientation, etc.).

Blum et al. [J.ACM' 13] to achieve scalability and practicality. Specifically, I am eager to investigate the effectiveness of applying GAN to generating a set of (accurate) candidates for synthetic data generation algorithm of Blum et al. Another possible line of research to adress this important problem, i.e., DP accurate synthetic data generation, is to formulate a dual constraints (utility & complexity) optimization problem over the set of variables in the data generation algorithm, and extend the methodology of the R^2DP framework, which is now used for a single constraint, to solve this problem. I argue that both ideas require rigorous analysis such as privacy analysis and empirical evaluations.

6.3 Privacy Preserving Distributed Computation

I am also interested in improving the existing privacy-preserving solutions in cloud and network security monitoring. In particular, I am interested in the following proposals.

- I am interested in adapting the idea of the multi-view paper to the multi-party scenario where several data owners are willing to share their traces to mitigate coordinated network reconnaissance by means of distributed (or inter-domain) audit.

- Popular approaches to differential privacy, such as the Laplace and exponential mechanisms, calibrate randomised smoothing through global sensitivity of the target non-private function. Bounding such sensitivity is often a prohibitively complex analytic calculation, especially over network data which has resulted in few solutions for limited applications, e.g., DP network trace analysis [SIGCOMM'10]. I argue that other definitions of DP, e.g., Random Differential Privacy (RDP) [Journal of Privacy and Confidentiality'12] or Rényi Differential Privacy [CSF'17] which rely on more practical sensitivity definitions, seem to be more appropriate under these scenarios. I am interested in applying these rigorous notions of privacy to the state-of-the-art security monitoring systems.

- Secure multiparty computation (SMC) is known as an effective tool in computing on private

data that was collected from many users. However, the common thread in all existing implementations of SMC is large scale computation, run by big organizations, on data that has been collected from many individual users. Recently, Mazloom et al. [CCS'18] established a new trade-off with privacy. Specifically, instead of claiming that the servers learn nothing about the input values (from SMC's output), their model claims that what they do learn from the computation preserves the differential privacy of the input. Leveraging this relaxation of the security model allows us to build a protocol that leaks some information in the form of access patterns to memory, while also providing a formal bound on what is learned from the leakage. On the other hand, this leakage results in a significant computation cost payoff in a broad class of computations such as histograms, PageRank and matrix factorization, which can be performed in common graph-parallel frameworks such as MapReduce or Pregel. I am interested in applying this new technology over more practical network/cloud auditing systems and IDS and evaluate the results of this system against other exiting frameworks.

Bibliography

[1] Trove page for hardy, g. h. (godfrey harold) (1877-1947).

[2] G. A. Aarons, A. E. Green, L. A. Palinkas, S. Self-Brown, D. J. Whitaker, J. R. Lutzker, J. F. Silovsky, D. B. Hecht, and M. J. Chaffin. Dynamic adaptation process to implement an evidence-based child maltreatment intervention. *Implementation Science*, 7(1):32, 2012.

[3] G. Acs, C. Castelluccia, and R. Chen. Differentially private histogram publishing through lossy compression. In *12th IEEE International Conference on Data Mining*, ICDM '12, pages 1–10, Brussels, Belgium, 2012.

[4] C. C. Aggarwal. Outlier analysis. In *Data mining*, pages 237–263. Springer, 2015.

[5] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 901–914, 2013.

[6] H. S. Asif, P. A. Papakonstantinou, and J. Vaidya. How to accurately and privately identify anomalies. In L. Cavallaro, J. Kinder, X. Wang, and J. Katz, editors, *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, pages 719–736. ACM, 2019.

[7] H. S. Asif, T. Talukdar, J. Vaidya, B. Shafiq, and N. R. Adam. Collaborative differentially private outlier detection for categorical data. In *2nd IEEE International Conference on Collaboration and Internet Computing, CIC 2016, Pittsburgh, PA, USA, November 1-3, 2016*, pages 92–101. IEEE Computer Society, 2016.

[8] H. S. Asif, T. Talukdar, J. Vaidya, B. Shafiq, and N. R. Adam. Differentially private outlier detection in a collaborative environment. *Int. J. Cooperative Inf. Syst.*, 27(3):1850005:1–1850005:36, 2018.

[9] B. Balle and Y. Wang. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *Proceedings of the 35th International Conference on Machine Learning*, ICML '18, pages 403–412, Stockholm, Sweden, 2018.

[10] B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar. Privacy, accuracy, and consistency too: A holistic solution to contingency table release. In *Proceedings of*

the 26th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS '07, pages 273–282, Beijing, China, 2007. ACM.

[11] M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In *Annual International Cryptology Conference, . , , Heidelberg*, pages 535–552. Springer, Berlin, 2007.

[12] D. P. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

[13] K. Bhaduri, M. D. Stefanski, and A. N. Srivastava. Privacy-preserving outlier detection through random nonlinear data distortion. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(1):260–272, Feb 2011.

[14] R. Bild, K. A. Kuhn, and F. Prasser. Safepub: A truthful data anonymization algorithm with strong privacy guarantees. *Proceedings on Privacy Enhancing Technologies*, 2018(1):67–87, 2018.

[15] P. Biler and A. Witkowski. Problems in mathematical analysis, 1990.

[16] V. Bindschaedler, P. Grubbs, D. Cash, T. Ristenpart, and V. Shmatikov. The tao of inference in privacy-protected databases. In *Proc. VLDB Endow*, pages 1715–1728, 11 (July 2018, 2018. 11.

[17] D. M. Bittner, A. D. Sarwate, and R. N. Wright. Using noisy binary search for differentially private anomaly detection. In I. Dinur, S. Dolev, and S. Lodha, editors, *Cyber Security Cryptography and Machine Learning - Second International Symposium, CSCML 2018, Beer Sheva, Israel, June 21-22, 2018, Proceedings*, volume 10879 of *Lecture Notes in Computer Science*, pages 20–37. Springer, 2018.

[18] A. Blum, K. Ligett, and A. Roth. A learning theory approach to non-interactive database privacy. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, STOC '08, pages 609–618, New York, NY, USA, 2008. ACM.

[19] J. Böhler, D. Bernau, and F. Kerschbaum. Privacy-preserving outlier detection for data streams. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 225–238. Springer, 2017.

[20] A. Boldyreva, N. Chenette, Y. Lee, and A. O'neill. Order-preserving symmetric encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques, . , , Heidelberg*, pages 224–241, Berlin, 2009. Springer.

[21] D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In *Theory of Cryptography Conference, . , , Heidelberg*, pages 253–273. Springer, Berlin, 2011.

[22] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Optimal geo-indistinguishable mechanisms for location privacy. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 251–262, 2014.

[23] T. Brekne. Årnes, a., & øslebø, a. (2005, may). In I. International, editor, *Anonymization of ip traffic monitoring data: Attacks on two prefix-preserving anonymization schemes and some proposed remedies*, pages 179–196. Springer, Berlin.

[24] T. a. Brekne. and andré årnes. *Circumventing IP-address pseudonymization. In Communications and Computer Networks, pp*, pages 43–48, 2005.

[25] H. Bremermann. Distributions, complex variables, and fourier transforms. 1965.

[26] H. Brenner and K. Nissim. Impossibility of differentially private universally optimal mechanisms. In *IEEE 51st Annual Symposium on Foundations of Computer Science*, FOCS '10, pages 71–80, Las Vegas, Nevada, USA, 2010.

[27] J. W. S. Brown, O. Ohrimenko, and R. Tamassia. Haze: Privacy-preserving real-time traffic statistics. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '13, pages 540–543, New York, NY, USA, 2013. ACM.

[28] M. G. Bulmer. *Principles of statistics*. Courier Corporation, 1979.

[29] M. Burkhart, D. Brauckhoff, M. May, and E. Boschi. The risk-utility tradeoff for ip address truncation. *In Proceedings of the*, 1:23–30, 2008.

[30] M. Burkhart, M. Strasser, D. Many, and X. Dimitropoulos. Sepia: Privacy-preserving aggregation of multi-domain network events and statistics. In *Proceedings of the 19th USENIX Conference on Security*, USENIX Security'10, pages 15–15, Berkeley, CA, USA, 2010. USENIX Association.

[31] M. Burkhart, M. Strasser, D. Many, and X. Dimitropoulos. Sepia: Privacy-preserving aggregation of multi-domain network events and statistics. In *Proceedings of USENIX Security Symposium*. 1, 2010.

[32] B. Caswell and J. Beale. *Snort 2.1 intrusion detection*. Elsevier, 2004.

[33] T. H. Chan, E. Shi, and D. Song. Private and continual release of statistics. *ACM Transactions Information System Security*, 14(3):26:1–26:24, 2011.

[34] Z. Chang, D. Xie, and F. L. O. ram. a dissection and experimental evaluation. In *Proceedings of the VLDB Endowment 9*, pages 1113–1124, no. 12, 2016.

[35] T. Chanyaswad, A. Dytso, H. V. Poor, and P. Mittal. MVG mechanism: Differential privacy under matrix-valued query. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, pages 230–246, Toronto, ON, Canada, 2018.

[36] T. Chanyaswad, C. Liu, and P. Mittal. Ron-gauss: Enhancing utility in non-interactive private data release. *PoPETs*, 2019(1):26–46, 2019.

[37] C. A. Charalambides. *Combinatorial Methods in Discrete Distributions (Wiley Series in Probability and Statistics)*, volume 600. Wiley-Interscience, New York, NY, USA, 2005.

[38] K. Chen and L. Liu. Privacy preserving data classification with rotation perturbation. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 4–pp. IEEE, 2005.

[39] J. E. Cohen, Y. Derriennic, and G. Zbaganu. Majorization, monotonicity of relative entropy, and stochastic matrices. *Contemporary Mathematics*, 149:251–251, 1993.

[40] M. G. T. command manual. *ee. lbl. gov/html/contrib/tcpdpriv. 0. txt.*, 1996, 1996.

[41] G. Cormode, C. M. Procopiuc, D. Srivastava, E. Shen, and T. Yu. Differentially private spatial decompositions. In *IEEE 28th International Conference on Data Engineering*, ICDE '12, pages 20–31, Washington, DC, USA, April 2012. IEEE Computer Society.

[42] S. E. Coull, F. Monrose, M. K. Reiter, and M. . Bailey. March). the challenges of effectively anonymizing network data. *In Conference For Homeland Security CATCH'0*, 9:230–236, 2009.

[43] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky. Searchable symmetric encryption: improved definitions and efficient constructions. *Journal of Computer Security*, 19(5):895–934, 2011.

[44] N. V. Dijkhuizen and J. V. D. Ham. A survey of network traffic anonymisation techniques and implementations. acm comput. *Surv.*, 51:3, May 2018.

[45] C. Dimitrakakis, B. Nelson, A. Mitrokotsa, and B. I. P. Rubinstein. Robust and private bayesian inference. In P. Auer, A. Clark, T. Zeugmann, and S. Zilles, editors, *Algorithmic Learning Theory*, pages 291–305, Cham, 2014. Springer International Publishing.

[46] B. Ding, M. Winslett, J. Han, and Z. Li. Differentially private data cubes: Optimizing noise sources and consistency. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, SIGMOD '11, pages 217–228, Athens, Greece, 2011.

[47] W. Ding, W. Yurcik, and X. Yin. Outsourcing internet security: Economic analysis of incentives for managed security service providers. In *International Workshop on Internet and Network Economics, . Heidelberg*, pages 947–958. Springer, Berlin, 2005.

[48] F. B. Durak, T. M. DuBuisson, and D. Cash. What else is revealed by order-revealing encryption? In *16). ACM, New York, NY, USA*, pages 1155–1166. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS, 2016.

[49] F. B. Durak, T. M. DuBuisson, and D. Cash. What else is revealed by order-revealing encryption? In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1155–1166, 2016.

[50] C. Dwork. Differential privacy: A survey of results. In *Theory and Applications of Models of Computation*, volume 4978, pages 1–19, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[51] C. Dwork. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation, . , , Heidelberg*, pages 1–19, Berlin, 2008. Springer.

[52] C. Dwork. Differential privacy: A survey of results. In M. Agrawal, D. Du, Z. Duan, and A. Li, editors, *Theory and Applications of Models of Computation*, pages 1–19, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[53] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *25th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[54] C. Dwork and J. Lei. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 371–380, 2009.

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

[56] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum. Differential privacy under continual observation. In *Proceedings of the 42rd ACM Symposium on Theory of Computing*, STOC '10, pages 715–724, New York, NY, USA, 2010. ACM.

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

[58] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, Aug. 2014.

[59] C. Dwork, G. N. Rothblum, and S. P. Vadhan. Boosting and differential privacy. In *Proceedings of the 51th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '10, pages 51–60, Las Vegas, Nevada, USA, Oct 2010.

[60] M. Dworkin. Recommendation for block cipher modes of operation: methods for format-preserving encryption. *NIST Special Publication*, 800, 2016.

[61] S. M. Erfani, Y. W. Law, S. Karunasekera, C. A. Leckie, and M. Palaniswami. Privacy-preserving collaborative anomaly detection for participatory sensing. In V. S. Tseng, T. B. Ho, Z.-H. Zhou, A. L. P. Chen, and H.-Y. Kao, editors, *Advances in Knowledge Discovery and Data Mining*, pages 581–593, Cham, 2014. Springer International Publishing.

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

[63] L. Fan and H. Jin. A practical framework for privacy-preserving data analytics. In *15). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland*, pages 311–321. Proceedings of the 24th International Conference on World Wide Web (WWW, 2015.

[64] T. Farah and L. Trajković. Anonym: A tool for anonymization of the internet traffic. In *2013 IEEE International Conference on Cybernetics (CYBCO)*, pages 261–266. IEEE, 2013.

[65] W. Feller. *An introduction to probability theory and its applications*, volume 2. John Wiley & Sons, 2008.

[66] M. Fisz. *Probability theory and mathematical statistics*, volume 3. 2018.

[67] M. Foukarakis, D. Antoniades, S. Antonatos, and E. P. Markatos. Flexible and high-performance anonymization of netflow records using anontool. In *2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops-SecureComm 2007*, pages 33–38. IEEE, 2007.

[68] M. Foukarakis, D. Antoniades, and M. Polychronakis. Deep packet anonymization. *In Proceedings of the Second European Workshop on System Security, pp*, pages 16–21, 2009.

[69] S. Gattani and T. E. Daniels. Reference models for network data anonymization. *In Proceedings of the*, 1:41–48, 2008.

[70] J. Gehrke, M. Hay, E. Lui, and R. Pass. Crowd-blending privacy. In *Annual Cryptology Conference, . , , Heidelberg*, pages 479–496. Springer, Berlin, 2012.

[71] Q. Geng, W. Ding, R. Guo, and S. Kumar. Optimal noise-adding mechanism in additive differential privacy. *CoRR*, abs/1809.10224, 2018.

[72] Q. Geng, P. Kairouz, S. Oh, and P. Viswanath. The staircase mechanism in differential privacy. *IEEE Journal of Selected Topics Signal Processing*, 9(7):1176–1184, 2015.

[73] Q. Geng and P. Viswanath. The optimal mechanism in differential privacy. In *2014 IEEE International Symposium on Information Theory*, pages 2371–2375, Honolulu, HI, USA, June 2014.

[74] Q. Geng and P. Viswanath. Optimal noise adding mechanisms for approximate differential privacy. *IEEE Transactions on Information Theory*, 62(2):952–969, Feb 2016.

[75] C. Gentry. Fully homomorphic encryption using ideal lattices. In *09). ACM, New York, NY, USA*, pages 169–178. Proceedings of the forty-first annual ACM symposium on Theory of computing (STOC, 2009.

[76] A. Ghosh, T. Roughgarden, and M. Sundararajan. Universally utility-maximizing privacy mechanisms. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, STOC '09, pages 351–360, New York, NY, USA, 2009. ACM.

[77] M. Gil. *On Rényi divergence measures for continuous alphabet sources*. PhD thesis, Citeseer, 2011.

[78] O. Goldreich. Secure multi-party computation. *Manuscript. Preliminary version :*, pages 86–97, 1998.

[79] O. Goldreich and R. Ostrovsky. Software protection and simulation on oblivious rams. *J. ACM*, 43(3):431–473, May 1996.

[80] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005.

[81] P. Grubbs, M.-S. Lacharite, B. Minaud, and K. G. Paterson. Learning to reconstruct: Statistical learning theory and encrypted database attacks. In I. S. on Security and, editor, *and Privacy (S&P) 2019 San Francisco,*. United States, May 2019.

[82] M. Gupte and M. Sundararajan. Universally optimal privacy mechanisms for minimax agents. In *Proceedings of the 29th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '10, pages 135–146, New York, NY, USA, 2010. ACM.

[83] P. Haag. *Watch your Flows with NfSen and NFDUMP*. In 50th RIPE Meeting, 2005.

[84] R. Hall, L. Wasserman, and A. Rinaldo. Random differential privacy. *Journal of Privacy and Confidentiality*, 4(2), 2013.

[85] M. Hardt, K. Ligett, and F. McSherry. A simple and practical algorithm for differentially private data release. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems*, NIPS '12, pages 2348–2356, Lake Tahoe,Nevada, USA, 2012.

[86] M. Hardt and G. N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *Proceedings of the 51th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '10, pages 61–70, Las Vegas, Nevada, USA, 2010.

[87] M. Hardt and K. Talwar. On the geometry of differential privacy. In *Proceedings of the 42rd ACM Symposium on Theory of Computing*, STOC '10, pages 705–714, New York, NY, USA, 2010. ACM.

[88] J. Hautakorpi and G. C. Gonzalez. Ip address distribution in middleboxes. *U.S. Patent Application No*, 12.

[89] M. Hay, C. Li, G. Miklau, and D. Jensen. Accurate estimation of the degree distribution of private networks. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, ICDM '09, pages 169–178, Washington, DC, USA, 2009. IEEE Computer Society.

[90] M. Hay, V. Rastogi, G. Miklau, and D. Suciu. Boosting the accuracy of differentially private histograms through consistency. *VLDB*, 3(1-2):1021–1032, Sept. 2010.

[91] A. Inan, M. Kantarcioglu, G. Ghinita, and E. Bertino. Private record matching using differential privacy. In *Proceedings of the 13th International Conference on Extending Database Technology*, pages 123–134, 2010.

[92] M. Jambunathan. Some properties of beta and gamma distributions. *The annals of mathematical statistics*, 25(2):401–405, 1954.

[93] J. L. W. V. Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta mathematica*, 30(1):175–193, 1906.

[94] Z. Ji, Z. C. Lipton, and C. Elkan. Differential privacy and machine learning: a survey and review. *arXiv preprint arXiv:1412.7584*, 2014.

[95] X. Jiang, Z. Ji, S. Wang, N. Mohammed, S. Cheng, and L. Ohno-Machado. Differential-private data publishing through component analysis. *Transactions Data Privacy*, 6(1):19–34, 2013.

[96] Z. Jorgensen, T. Yu, and G. Cormode. Conservative or liberal? personalized differential privacy. In *2015 IEEE 31St international conference on data engineering*, pages 1023–1034. IEEE, 2015.

[97] S. P. Kasiviswanathan, K. Nissim, S. Raskhodnikova, and A. Smith. Analyzing graphs with node differential privacy. In A. Sahai, editor, *Theory of Cryptography*, pages 457–476, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[98] G. Kellaris, G. Kollios, K. Nissim, and A. O'neill. Generic attacks on secure outsourced databases. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1329–1340, 2016.

[99] R. B. Kellogg. CRC standard mathematical tables and formulae. *SIAM Review*, 38(4):691–692, 1996.

[100] M. Kim. Anomaly detection, 2020. `https://www.mathworks.com/matlabcentral/fileexchange/39593-anomaly-detection`.

[101] R. Kohavi. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD' 96, pages 202–207. AAAI Press, 1996.

[102] F. Koufogiannis, S. Han, and G. J. Pappas. Optimality of the laplace mechanism in differential privacy. *arXiv preprint arXiv:1504.00065*, 2015.

[103] D. Koukis, S. Antonatos, D. Antoniades, E. P. Markatos, and P. Trimintzios. A generic anonymization framework for network traffic. In *2006 IEEE International Conference on Communications*, pages 2302–2309, IEEE, 2006. vol. 5.

[104] E. Kushilevitz and R. Ostrovsky. Replication is not needed: Single database, computationally-private information retrieval. *In Foundations of Computer Science*, 1997:364–373, 1997.

[105] J. Le Ny and G. J. Pappas. Differentially private filtering. *IEEE Transactions on Automatic Control*, 59(2):341–354, 2014.

[106] S. Lee, E. L. Wong, D. Goel, M. Dahlin, and V. Shmatikov. πbox: A platform for privacy-preserving apps. In *Proceedings of the 10th {USENIX} Symposium on Networked Systems Design and Implementation*, {NSDI} '13, pages 501–514, Lombard, IL, USA, 2013.

[107] J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection, Jun 2014. `http://snap.stanford.edu/data`.

[108] C. Li, M. Hay, G. Miklau, and Y. Wang. A data- and workload-aware algorithm for range queries under differential privacy. *VLDB*, 7(5):341–352, Jan. 2014.

[109] C. Li, M. Hay, V. Rastogi, G. Miklau, and A. McGregor. Optimizing linear counting queries under differential privacy. In *Proceedings of the 29th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '10, pages 123–134, Indianapolis, Indiana, USA, 2010. ACM.

[110] C. Li, M. Hay, V. Rastogi, G. Miklau, and A. McGregor. Optimizing linear counting queries under differential privacy. In *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 123–134, 2010.

[111] C. Li, G. Miklau, M. Hay, A. McGregor, and V. Rastogi. The matrix mechanism: optimizing linear counting queries under differential privacy. *VLDB J.*, 24(6):757–781, 2015.

[112] L. Li, L. Huang, W. Yang, X. Yao, and A. Liu. Privacy-preserving LOF outlier detection. *Knowl. Inf. Syst.*, 42(3):579–597, 2015.

[113] N. Li, W. Qardaji, and D. Su. Provably private data anonymization: Or, k-anonymity meets differential privacy. *CoRR. abs/*, 1101:2604, 2011.

[114] N. Li, W. Qardaji, and D. Su. On sampling. In *12). ACM, New York, NY, USA*, pages 32–33, and differential privacy or, k-anonymization meets differential privacy. In Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security (ASIACCS, 2012. anonymization.

[115] Y. Li, A. Slagell, K. Luo, and W. Yurcik. A combined conversion and anonymization tool for processing NetFlows for security, CANINE, 2005.

[116] D. Liu and S. Wang. Nonlinear order preserving index for encrypted database query in service cloud environments. *Concurrency and Computation: Practice and Experience*, 25(13):1967–1984, 2013.

[117] K. Liu, C. Giannella, and H. Kargupta. An attacker's view of distance preserving maps for privacy preserving data mining. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 297–308. Springer, 2006.

[118] T. D. Luong and T. B. Ho. A distributed solution for privacy preserving outlier detection. In *Third International Conference on Knowledge and Systems Engineering, KSE 2011, Hanoi, Vietnam, October 14-17, 2011*, pages 26–31. IEEE Computer Society, 2011.

[119] L. Lyu, Y. W. Law, S. M. Erfani, C. Leckie, and M. Palaniswami. An improved scheme for privacy-preserving collaborative anomaly detection. In *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pages 1–6. IEEE, 2016.

[120] M. Lyu, D. Su, and N. Li. Understanding the sparse vector technique for differential privacy. *PVLDB*, 10(6):637–648, 2017.

[121] T. Mayberry, E.-O. Blass, and A. H. Chan. *Efficient Private File Retrieval by Combining ORAM and PIR*. In NDSS, 2014.

[122] R. Mayer, M. Hittmeir, and A. Ekelhart. Privacy-preserving anomaly detection using synthetic data. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 195–207. Springer, 2020.

[123] F. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, SIGMOD '09, pages 19–30, Rhode Island, USA, 2009. ACM.

[124] F. McSherry and R. Mahajan. Differentially-private network trace analysis. *SIGCOMM Comput. Commun. Rev.*, 40(4):123–134, Aug. 2010.

[125] F. McSherry and R. Mahajan. Differentially-private network trace analysis. *In ACM SIGCOMM Computer Communication Review*, 40(4):123–134, 2010.

[126] F. McSherry and I. Mironov. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, page 627–636, New York, NY, USA, 2009. Association for Computing Machinery.

[127] G. Minshall. Tcpdpriv. ee. lbl. gov/html/contrib/tcpdpriv. html, 1997.

[128] J. Mirkovic. Privacy-safe network trace sharing via secure queries. In *08). ACM, New York, NY, USA*, pages 3–10. Proceedings of the 1st ACM workshop on Network data anonymization (NDA, 2008.

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

[130] P. Mittal, V. Paxson, R. Sommer, and M. Winterrowd. *Securing Mediated Trace Access Using Black-box Permutation Analysis*. In HotNets, 2009.

[131] K. Mivule and B. Anderson. A study of usability-aware network trace anonymization. *In*, 2015:1293–1304, 2015.

[132] J. C. Mogul and M. Arlitt. Sc2d: an alternative to trace anonymization. *In Proceedings of the*, 2006:323–328, 2006.

[133] M. Mohammady, L. Wang, Y. Hong, H. Louafi, M. Pourzandi, and M. Debbabi. Preserving both privacy and utility in network trace anonymization. In *18). ACM, New York, NY, USA*, pages 459–474. Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS, 2018.

[134] M. Mohammady, L. Wang, Y. Hong, H. Louafi, M. Pourzandi, and M. Debbabi. Preserving both privacy and utility in network trace anonymization. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, pages 459–474, New York, NY, USA, 2018. ACM.

[135] D. Moore, K. Keys, R. Koga, E. Lagache, and K. C. Claffy. The coralreef software suite as a tool for system and network administrators. In *01). USENIX Association, Berkeley, CA, USA*, pages 133–144. Proceedings of the 15th USENIX conference on System administration (LISA, 2001.

[136] M. Naveed, S. Kamara, and C. V. Wright. Inference attacks on property-preserving encrypted databases. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 644–655. ACM, 2015.

[137] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi. Dïot: A federated self-learning anomaly detection system for iot. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 756–767. IEEE, 2019.

[138] A. Nikolov, K. Talwar, and L. Zhang. The geometry of differential privacy: the sparse and approximate cases. In *Symposium on Theory of Computing Conference*, STOC '13, pages 351–360, Palo Alto, CA, USA, 2013.

[139] K. Nissim, S. Raskhodnikova, and A. D. Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, STOC '07, pages 75–84, San Diego, California, USA, 2007.

[140] J. L. Ny and M. Mohammady. Differentially private MIMO filtering for event streams. *IEEE Transactions on Automatic Control*, 63(1):145–157, Jan 2018.

[141] R. Okada, K. Fukuchi, and J. Sakuma. Differentially private analysis of outliers. In A. Appice, P. P. Rodrigues, V. S. Costa, J. Gama, A. Jorge, and C. Soares, editors, *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part II*, volume 9285 of *Lecture Notes in Computer Science*, pages 458–473. Springer, 2015.

[142] R. Pang, M. Allman, V. Paxson, and J. Lee. The devil and packet trace anonymization. *ACM SIGCOMM Computer Communication Review*, 36(1):29–38, 2006.

[143] R. Pang and V. Paxson. A high-level programming environment for packet trace anonymization and transformation. In *03). ACM, New York, NY, USA*, pages 339–351, technologies, architectures, and protocols for computer communications (SIGCOMM, 2003. Proceedings of the 2003 conference on Applications.

[144] H. H. Panjer. Recursive evaluation of a family of compound distributions. *ASTIN Bulletin*, 12(1):22–26, 1981.

[145] V. Rastogi and S. Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, pages 735–746, Indianapolis, Indiana, USA, 2010.

[146] A. Rényi et al. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California, 1961.

[147] B. F. Ribeiro, W. Chen, G. Miklau, and D. F. Towsley. *Analyzing Privacy in Enterprise Packet Trace Anonymization*. In NDSS, 2008.

[148] D. Riboni, A. Villani, D. Vitali, C. Bettini, and L. V. Mancini. Obfuscation of sensitive data in network flows. *In INFOCOM*, 2012:2372–2380, 2012.

[149] D. Riboni, A. Villani, D. Vitali, C. Bettini, and L. V. Mancini. Obfuscation of sensitive data for incremental release of network flows. *IEEE/ACM Transactions on Networking*, 23(2):672–686, 2015.

[150] S. Rolweicz and E. Bedarczuk. *Functional Analysis and Control Theory: Linear Systems*. Kluwer Academic Publishers, USA, 1986.

[151] M. Roughan. Public review for the devil and packet trace anonymization. sigcomm comput. *Commun. Rev*, 36(1):27–28, January 2006.

[152] B. I. P. Rubinstein and F. Aldà. Pain-free random differential privacy with sensitivity sampling. volume 70 of *Proceedings of Machine Learning Research*, pages 2950–2959, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[153] A. J. Slagell, K. Lakkaraju, and K. Luo. Flaim: A multi-level anonymization framework for computer and network logs. *In LISA*, 6:3–8, 2006.

[154] A. J. Slagell, K. Lakkaraju, and K. Luo. Flaim: A multi-level anonymization framework for computer and network logs. *In LISA*, 6:3–8, 2006.

[155] A. J. Slagell, Y. Li, and K. Luo. Sharing network logs for computer forensics: A new tool for the anonymization of netflow records. In *Workshop of the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks*, pages 37–42, IEEE, 2005. 2005.

[156] D. X. Song, D. Wagner, and A. Perrig. Practical techniques for searches on encrypted data. *In Security and Privacy*, 2000:44–55, 2000.

[157] E. Stefanov, M. Van Dijk, E. Shi, C. Fletcher, L. Ren, X. Yu, and S. Devadas. November. *Path ORAM: an extremely simple oblivious RAM protocol*, 20:299–310, 2013.

[158] S. J. Stolfo, Wei Fan, Wenke Lee, A. Prodromidis, and P. K. Chan. Cost-based modeling for fraud and intrusion detection: results from the jam project. In *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*, volume 2, pages 130–144 vol.2, Jan 2000.

[159] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, and Y. Zhou. A hybrid approach to privacy-preserving federated learning. In L. Cavallaro, J. Kinder, S. Afroz, B. Biggio, N. Carlini, Y. Elovici, and A. Shabtai, editors, *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2019, London, UK, November 15, 2019*, pages 1–11. ACM, 2019.

[160] J. Vaidya and C. Clifton. Privacy-preserving outlier detection. In *Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004), 1-4 November 2004, Brighton, UK*, pages 233–240. IEEE Computer Society, 2004.

[161] J. Vaidya, B. Shafiq, A. Basu, and Y. Hong. Differentially private naive bayes classification. In *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 01*, WI-IAT '13, page 571–576, USA, 2013. IEEE Computer Society.

[162] J. Vaidya, B. Shafiq, A. Basu, and Y. Hong. Differentially private naive bayes classification. In *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 1, pages 571–576, 2013.

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

[164] X. S. Wang, Y. Huang, T. H. H. Chan, A. Shelat, and E. Shi. Scoram: oblivious ram for secure computation. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 191–202. ACM, 2014.

[165] Y. Wang, Z. Huang, S. Mitra, and G. E. Dullerud. Entropy-minimizing mechanism for differential privacy of discrete-time linear feedback systems. In *53rd IEEE Conference on Decision and Control*, pages 2130–2135, Dec 2014.

[166] Y. Wang, Z. Huang, S. Mitra, and G. E. Dullerud. Entropy-minimizing mechanism for differential privacy of discrete-time linear feedback systems. In *53rd IEEE Conference on Decision and Control*, pages 2130–2135, Dec 2014.

[167] Y. Wang, X. Wu, and D. Hu. Using randomized response for differential privacy preserving data collection. In *Proceedings of EDBT/ICDT Workshops Joint Conference*, EDBT/ICDT '16, Bordeaux, France, 2016.

[168] C. R. Wren, Y. A. Ivanov, D. Leigh, and J. Westhues. The merl motion detector dataset. In *Proceedings of the 2007 Workshop on Massive Datasets*, MD '07, pages 10–14, New York, NY, USA, 2007. ACM.

[169] X. Xiao, G. Wang, and J. Gehrke. Differential privacy via wavelet transforms. *IEEE Transactions on knowledge and data engineering*, 23(8):1200–1214, 2010.

[170] X. Xiao, G. Wang, and J. Gehrke. Differential privacy via wavelet transforms. *IEEE Transactions on Knowledge and Data Engineering*, 23(8):1200–1214, Aug. 2011.

[171] C. Xu, J. Ren, Y. Zhang, Z. Qin, and K. Ren. Dppro: Differentially private high-dimensional data release via random projection. *IEEE Transactions Information Forensics and Security*, 12(12):3081–3093, 2017.

[172] J. Xu, J. Fan, M. H. Ammar, and S. B. Moon. Prefix-preserving ip address anonymization: Measurement-based security evaluation and a new cryptography-based scheme. In *10th IEEE International Conference on Network Protocols*, pages 280–289, IEEE, 2002. 2002. Proceedings.

[173] J. Xu, J. Fan, M. H. Ammar, and S. B. Moon. Prefix-preserving ip address anonymization: Measurement-based security evaluation and a new cryptography-based scheme. In *10th IEEE International Conference on Network Protocols*, pages 280–289, IEEE, 2002. 2002. Proceedings.

[174] A. Xue, X. Duan, H. Ma, W. Chen, and S. Ju. Privacy preserving spatial outlier detection. In *Proceedings of the 9th International Conference for Young Computer Scientists, ICYCS 2008, Zhang Jia Jie, Hunan, China, November 18-21, 2008*, pages 714–719. IEEE Computer Society, 2008.

[175] A. C.-C. Yao. How to generate and exchange secrets. *In Foundations of Computer Science*, 1986:162–167, 1986.

[176] T.-F. Yen, X. Huang, F. Monrose, and M. K. Reiter. Browser fingerprinting from coarse traffic summaries: Techniques and implications. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, . , , Heidelberg*, pages 157–175, Berlin, 2009. Springer.

[177] G. Yuan, Z. Zhang, M. Winslett, X. Xiao, Y. Yang, and Z. Hao. Low-rank mechanism: Optimizing batch queries under differential privacy. *PVLDB*, 5(11):1352–1363, 2012.

[178] G. Yuan, Z. Zhang, M. Winslett, X. Xiao, Y. Yang, and Z. Hao. Optimizing batch linear queries under exact and approximate differential privacy. *ACM Transactions Database Systems*, 40(2):11:1–11:47, 2015.

[179] W. Yurcik, C. Woolam, G. Hellings, L. Khan, and B. Thuraisingham. Scrub-tcpdump: A multi-level packet anonymizer demonstrating privacy/analysis tradeoffs. In *2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops-SecureComm 2007*, pages 49–56. IEEE, 2007.

[180] Q. Zhang and X. Li. An ip address anonymization scheme with multiple access levels. In *International Conference on Information Networking, . , , Heidelberg*, pages 793–802, Berlin, 2006. Springer.

[181] X. Zhang, R. Chen, J. Xu, X. Meng, and Y. Xie. Towards accurate histogram publication under differential privacy. In *Proceedings of the SIAM International Conference on Data Mining*, SDM '14, pages 587–595, Philadelphia, Pennsylvania, USA, 2014.

[182] Z. Zhang, B. I. Rubinstein, and C. Dimitrakakis. On the differential privacy of bayesian inference. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[183] T. Zhu, G. Li, W. Zhou, and S. Y. Philip. Differentially private data publishing and analysis: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 29(8):1619–1638, 2017.