Thesis for the Degree of Doctor of Engineering

Dynamic Enforcement of Differential Privacy

Hamid Ebadi

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
Göteborg, Sweden 2018

Dynamic Enforcement of Differential Privacy
Hamid Ebadi
ISBN 978-91-7597-685-3

© 2018 Hamid Ebadi

Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr 4366
ISSN ISSN 0346-718X

Technical Report 153D
Department of Computer Science and Engineering
Software Technology

Department of Computer Science and Engineering
Chalmers University of Technology SE-412 96 Göteborg
Sweden
Telephone +46 (0)31-772 1000

Printed at Chalmers
Göteborg, Sweden 2018

Abstract

With recent privacy failures in the release of personal data, differential privacy received considerable attention in the research community. This mathematical concept, despite its young age (Dwork et al., 2006), has grabbed the attention of many researchers for its robustness against identification of individuals even in presence of background information. Besides that, its flexible definition makes it compatible with different data sources, data mining algorithms and data release models. Its compositionality properties facilitate design of "differential privacy aware" programming languages and frameworks that empower non-experts to construct complex data mining analyses with proven differential privacy guarantees. The goal of this research is to introduce new (and improve the current) differential privacy backed frameworks, prominent both in utility and flexibility of use. We study dynamic enforcement of differential privacy both in the *centralised model* in which a trusted curator process data stored in a centralised database and the *local model* with no trust on the third party.

For the centralised model the thesis mostly focuses on the privacy impact of the basic building blocks used in these frameworks, proving correctness of the system built upon them. With respect to accuracy, we present *"personalised differential privacy"* as an improved method of enforcing privacy that provides better data utilisation and other benefits. In this setting, individuals take control of their privacy requirements rather than being seen as a part of a database. As a result, they can opt-in to a database with their expected privacy level and optionally opt-out later. We further study the privacy implication of other building blocks such as different kinds of sampling and partitioning.

For the local model we propose a general framework in which the users can verify the recieved analyses and with a flexible policy express their privacy preference in different forms such as enforcing their personalised privacy budget.

Acknowledgements

Thanks to my families for their unconditional love. Thanks to my supervisor, David Sands, for his patience, true kindness and his support. Thanks to Gerardo Schneider and Thibaud Antignac as my co-supervisors and Andrei Sabelfeld as my examiner for their help and guidance throughout my study. And finally to my friends in Sweden for the great times that we have spent together and all my friends at Chalmers for all the inspiring discussions during fika times.

Funding

This work was partially supported by a grant from the Swedish Foundation for Strategic Research (SSF).

Table of Contents

1 Introduction **1**
- 1 Why Does Privacy Matter? . . . 2
- 2 Differential Privacy . . . 5
- 3 Variants of Differential Privacy . . . 8
- 4 Scientific Methods and Goals . . . 10
- 5 Differential Privacy, Tools and Methods . . . 11
- 6 Contributions . . . 14
  - 6.1 Paper I : Featherweight PINQ . . . 15
  - 6.2 Paper II : Differential Privacy, Now it is Getting Personal 17
  - 6.3 Paper III : Sampling and Partitioning for Differential Privacy . . . 18
  - 6.4 Paper IV : PreTPost: A Transparent, User Verifiable, Local Differential Privacy Framework . . . 19
  - 6.5 Paper V : Design and Use of PreTPost Framework . . 19
  - 6.6 Personal Contributions . . . 20

2 Paper I : Featherweight PINQ **25**
- 1 Introduction . . . 30
- 2 PINQ . . . 32
- 3 Idealised Program . . . 33
- 4 Featherweight PINQ . . . 36
  - 4.1 The Protected System . . . 36
  - 4.2 The Featherweight PINQ Transition System . . . 36
- 5 Differential Privacy for Featherweight PINQ . . . 40
  - 5.1 Trace semantics . . . 41
- 6 Related Work . . . 43
- 7 Limitation and Extension . . . 44
- 8 Conclusion . . . 45
- 1 Proof of Theorem . . . 47

3 Paper II : Differential Privacy: Now it's Getting Personal **49**
- 1 Introduction . . . 52

2 Differential Privacy 54
3 Personalised Differential Privacy 56
4 ProPer: Provenance for Personalised Privacy 58
4.1 Overview of ProPer 59
4.2 Preliminary Definitions and Notation 60
4.3 Provenance Tracing 62
4.4 The System Model 64
4.5 Trace semantics 68
5 ProPer Provides $\mathscr{E}$-Differential Privacy 69
6 Implementation and Experimental Results 71
6.1 Description of the Tool 72
6.2 Example 73
6.3 Experimental results 75
6.4 Limitations 75
7 Related work 76
8 Conclusion 79

4 Paper III : Sampling and Partitioning for Differential Privacy 87
1 Introduction 89
2 Preliminaries 90
3 Sampling in PINQ 93
4 Uniform Sampling and Partitioning 95
4.1 From Deterministic to Probabilistic Transformations . 96
4.2 Uniform (Fixed Size) Sampling Without Replacement 99
4.3 Uniform (Fixed Size) Sampling with Replacement . . . 101
4.4 Fraction Sampling 101
4.5 Bernoulli Sampling 103
5 Related work 103
6 Conclusion 104
1.1 Proof of Concept to Demonstrate the PINQ's Weakness 107
1.2 Probabilistic Stability of Uniform Sampling (Without Replacement) 109
1.3 Probabilistic Stability of Uniform Partitioning (Without Replacement) 110
1.4 Recursive (Fixed Sized) Uniform Sampling (Without Replacement) 111

5 Paper IV : PreTPost: A Transparent, User Verifiable, Local Differential Privacy Framework 113
1 Introduction 116
2 Foundations 120
2.1 Differential Privacy 120
2.2 Composition Principles 121

2.3 Randomised Response 123
2.4 Personalised Privacy 124
3 Framework 125
3.1 The Aggregator Side 126
3.2 The User Side 127
3.3 Modelling Different Policies 129
3.4 Implementation 131
3.5 Deployment 132
4 Case Studies 133
5 Utility Issues 137
6 Related Work 138
7 Future Work 139
8 Conclusion 140

6 Paper V :
Design and Use of PreTPost Framework **147**
1 Introduction 149
2 Communication Between Curator and Users 151
3 Query Construction in the Curator Side 152
3.1 Pre-processing 155
3.2 Randomised Transformation 156
3.3 Post-processing 157
3.4 Query Transmission and Data Collection 158
4 Query Execution on the User Side 159
4.1 The Public Policy 159
4.2 Private Policy 161
4.3 Randomised Transformation and Domain Enforcement 161
5 Isolation 162
6 Experiment 163
7 Future Work 164
8 Conclusion 164

Introduction

Personal information is collected starting from morning; a transportation record is stored in a database when you use your contact-less bus card. Data is collected as the bus passes through mobile phone cells, when you are busy checking your emails or when you interact with others via social networks. Before you declare the time and your attendance in the office using biometric methods, you might use your credit card to buy a coffee from a shop protected by a CCTV camera. Different kinds of information are collected and stored in different databases, possibly in different geographical locations. Some of this data is legitimately collected to be used for strict and predefined purposes. In the above scenario, your salary or your transportation cost is calculated from the collected data.

Besides the desirable and legitimate processing of data, each service provider involved in this scenario can benefit from collected information in different ways. It is common for companies to use this data to learn about their customers in order to predict their behaviours and improve the service. As an example, the transportation company can change the number of buses in different areas, adjust the trip frequency according to the time of day, or place more stations within areas with higher traveller density.

Using personal data for such purposes seems not only benign but also desirable; however, setting a proper border on how data should be handled is a question that needs more scrutiny. Knowing that an analysis is constructed from individuals' personal information confronts us with the crucial question: is it possible for someone to learn about an individual by looking at the results of statistical analyses. Back to our example, looking at the schedule and frequency of buses, can travel information of an individual be inferred?

Even though the leakage in this specific case seems to be insignificant, how can this leakage be quantified? What if one bus stop is only used by one individual? While using the valuable information stored in a database is tempting, this information has the most value for those who own it. Any mistake or mishandling of these sensitive private data can easily cause chaos.

Investigating how much information can be extracted from the result of a statistical analysis helps us to identify possible threats, and to provide better assurance for users' privacy.

1 Why Does Privacy Matter?

Our personal information is valuable, but privacy, like security, becomes an important issue when a breach occurs. Considering websites and businesses that provide free services, solely by trading users' personal information, we see the immediate value that our personal information has. Unfortunately, with the current state of Internet and society, where people and Internet do not forget, the consequences of privacy violation in the long term are not rectifiable.

No one can quantify the physical or emotional harm resulting from a privacy breach since the type of possible harm, its duration and its effect on different people is not completely known. Data revealed as a result of a privacy breach may persist as long as the lifetime of its owner or even longer in the case of Genomic data. This explains why possible harm caused by privacy breaches should not be underestimated. While eating habits and the amount of alcohol an individual consumes, may be interesting for an advertising company, it is important to remember that the advertisement companies are not the only people who can take advantage of this information. This information can likewise be used in the process of recruitment for your next job, to effect the outcome of university selection, or be used by health insurance companies.

Similarly, while the leakage of a database that keeps individual coffee orders may at first sound insignificant, it reveals the person's location, or in other words, his absence from his office during working hours. The number of coffees that the person bought can reveal whether he was alone or accompanied by another person. Presence of two persons in the coffee shop at the same time may disclose an affair, special communication or friendship between them. Knowing the political orientation of one side may reveal more about the purpose of the meeting. Progress in science may reveal a pattern between coffee drinking and a mental/physical problem which gives new meanings and dimensions to this plain information record. As it can be seen, any kind of information that individually looks harmless can be abused if it is linked (possibly in the future) with meaningful background information. The harm of a privacy violation may go beyond the individual that is the subject to the privacy breach or even beyond their lifetime. In 1951 after Henrietta Lacks died from aggressive form of cervical cancer, her cells (also known as HeLa cells) were cultivated in a laboratory. Progress in genome sequencing suddenly introduced a new privacy threat to her family as HeLa cells' genome was published on Internet [40]. The genomes have much similarity with genes that she passed on to her children and grandchildren living today. Hid-

den in the sequence is potential biomedical information about Henrietta's descendants, such as their risk for getting Alzheimer's disease or other kinds of disorders. Recent studies [29, 28] have shown that social media contents can be used to infer other information that are not voluntarily expressed such as age, sex, political view, intelligence, sexual orientation and preference.

Privacy is a broad topic; in this thesis we focus on privacy in statistical databases. Since privacy is better understood when the violations are investigated, in the next section we review some failures caused by careless releases of databases.

Failures of Database Anonymisation If there is no privacy concern, public interest encourages information disclosure as a way to learn more about behaviours and common patterns. To achieve privacy, anonymisation (de-identification) – that is the process of removing personally identifiable information (PII) from raw data – is often used. A *quasi-identifier* is the unique key (such as social security number[1]) used to identify a person in a database. While the idea of removing quasi-identifiers from database for anonymisation seems to be promising, recent efforts in data anonymisation have ended up in scandals. In what follows list some of the most well-known privacy failures of anonymisation efforts is listed.

AOL Search Query Data Scandal When AOL released a huge dataset of search queries belonging to 650,000 users, the company claimed that they had applied anonymisation thoroughly as the IP addresses were removed and user IDs were replaced with random numbers. The random numbers avoid identification of individuals while at the same time allowing researchers to correlate different search queries that belong to individuals. While preliminary thought suggests that these data are random terms issued by random people, looking at search queries one can notice a person's feelings and thoughts in different social or emotional conditions. Using these, it is possible to identify a person who wants to commit suicide, someone who looks for a restaurant nearby and someone looking for medication by searching symptoms.

These terms can easily reveal someone's identity when a user searches for a friend's name, a nearby location or a rare disease. In these cases, the searched terms can be linked together to single out and expose the faces behind them [4]. Once the person is identified, her/his other queries can reveal his/her deepest feelings, passions or secrets. To show the privacy risk, the New York Times revealed the face behind the user No. 4417749 who searched for "numb fingers", "60 single men", "homes sold in shadow lake subdivision gwinnett county georgia" , "landscapers in Lilburn, Ga," and "dog that urinates on everything.".

[1] Similar to the personal number in Sweden

Massachusetts Group Insurance Commission In another privacy incident, the "Massachusetts Group Insurance Commission" released hospital visit information for state employees after removing social security numbers and addresses. William Weld, the Governor of Massachusetts claimed that the anonymisation of the information protects the privacy of patients. Sweeney [42] purchased the voter registration list for Cambridge Massachusetts city and linked these two databases to find the governor's health record. She narrowed down the possible records that may belong to William Weld to three, using only birthday and gender. Finally, using the five digits ZIP code, Weld's medical record was uniquely identified revealing information about his ethnicity, visit dates, diagnosis, procedures, charges and medications. While name and social security number are clearly considered the key to identify a person, Sweeney's research concluded that combination of ZIP code, birthday and sex can alternatively be used to uniquely identify 87 percent of the population in the US.

Netflix Competition The process of correlating and linking data to individuals is surprisingly effective knowing the person's preferences or interests. User preference, like ratings given to a movie, that has never seriously been considered personally identifiable information, is shown to be informative enough for re-identification purposes. When Netflix announced a prize for an alternative algorithm that can predict user ratings for movies by mining current users' ratings, a real dataset of users' ratings was provided to let the contesters train, test and measure the quality of their algorithms.

The dataset had 100,480,507 ratings that 480,189 users gave to 17,770 movies. To anonymise the dataset, user IDs were replaced, some ratings were randomly changed, added or deleted and some rating dates were also modified. Even with this level of anonymisation, Narayanan and Shmatikov [33] demonstrated that movie rental details (time and the given rating) of three movies are enough to successfully link an individual to his records in the database. They compared the dataset against publicly available IMDB ratings as the source of background information. As a result of their research, one can conclude that the combination of spatial and high dimensional attributes can make re-identification easy.

As the result of these privacy scandals, AOL took down the user search database several hours later. Under the pressures of a lawsuit and Federal Trade Commission questioning, Netflix also decided to shut down the contest. These countermeasures, as usual, were not effective as both databases have been mirrored immediately and distributed over the Internet.

Background knowledge and cross-correlation with public databases is a common way to re-identify individuals when obvious quasi-identifiers are removed. Re-identification is done when the result of linking our data

with a background database leaves out few possible individuals for each anonymised record. Sweeney [42] proposed the *k-anonymity* property for the data. In k-anonymity the data provider (by manipulating data) ensures that each individual's information is indistinguishable from at least $k-1$ other individuals present in the database. While it is not difficult to see why k-anonymity totally fails in high dimensional databases, knowing possible attributes, even though not precisely, can still be informative. As an example, a dataset that has 3-anonymity with respect to the attribute disease may reveal that one person has HIV, syphilis or chlamydia. K-anonymity also fails completely when the attacker is in possession of additional knowledge. This background information can be a database that is released with a k-anonymity guarantee which shows k-anonymity does not compose with itself.

As explained earlier in this section, in non-interactive data disclosure, sanitized data is released after applying some anonymisations (also known as de-identification) once and for all. Removing obvious identifiers, sub-sampling and perturbing values are used to construct the "synthetic data". Unfortunately, the data released in non-interactive methods is only suitable for the particular class of data analyses that it is released for. In the next section, the possibility of a more fine-grained data release method, flexible enough to be used in different scenarios that supports interactive systems, will be discussed.

2 Differential Privacy

The consequences of failures in anonymisation show the challenge of statistical data disclosure. It seems the perfect privacy scenario from every individual's perspective is achieved when the individual opts-out from the analysis and removes their record from the database. Privacy for everyone means that all records in the database have to be removed which results in an empty database with no analysis. This is in line with another expectation of privacy that the access to a database should not reveal anything about an individual that cannot be gained without the access. Dwork [8] shows the absolute disclosure prevention is impossible in presence of auxiliary information that may be totally irrelevant to the database. The impossibility result can be demonstrated with a simple scenario. Assume the money that Zlatan Ibrahimović spent for alcohol in last midsommarfest[2] is considered to be secret. If auxiliary information reveals that the money could feed 100 people in Fiji for a week, access to the database of expenses in Fiji reveals the sum of the purchases, regardless of Zlatan's presence in that database. The fact that we cannot protect his privacy is inevitable and leads to a new formulation of privacy known as *differential privacy* that makes no assumption about the auxiliary information. This means even a computationally powerful ad-

[2] midsommarfest is a Swedish celebration held to welcome summer

versary that has information about all records in the database except the target individual is not able to learn more from the individual beyond the specified limit.

Other methods of anonymisation usually suffer from two problems. The first problem arises in corner cases in which a few individuals are effectively singled-out in analyses. As the number of individuals participating in an analysis decreases the effect of each individual and subsequently the information leak about those individuals increases. Each independent analysis that is done independently leaks information too. One major difficulty is quantifying the information leakage for each analysis and the total information leakage caused by composition of several analyses.

Having all this in mind, Dwork et al. [10] formulated differential privacy as indistinguishability of outcomes when an analysis is performed on two similar databases. The *hamming distance* is commonly used as a similarity measure for databases and closeness of probability of outcomes measures the indistinguishability. A computation Q over these datasets provides ϵ-differential privacy if it keeps the ratio of probabilities (A and B as shown in Fig 1) of observing any outcome lower than e^{ϵ} when it runs on two neighbouring databases D and D' (with hamming distance 1). More precisely the definition is:

Definition 1. *A randomized function Q gives ϵ-differential privacy if for all $S \subseteq Range(Q)$ and all D and D' where they are neighbours, we have:* $|\frac{Pr[Q(D)\in S]}{Pr[Q(D')\in S]}| \leq e^{\epsilon}$

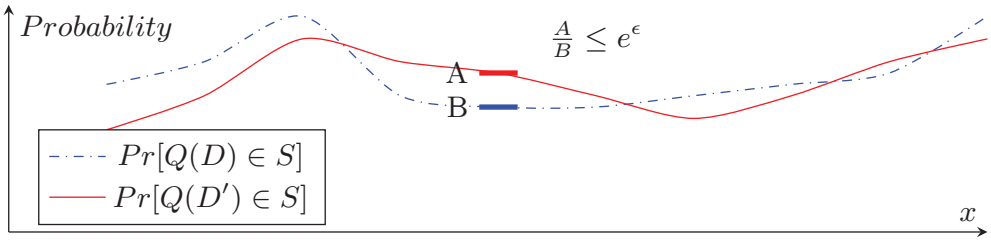

Fig. 1. Outcome Probability

Properties As it can be realised from the definition, differential privacy is flexible and not restricted to a database schema or the type of output. The basic principles used to construct analyses over the Netflix tabular database can be used to query more sophisticated data structures like graphs used to represent interactions in social networks. Differential privacy not only protects the value of data points but also protects the existence of individuals in a database. Imprecise responses give differential privacy the desirable deniability feature even with possession of everyone else's personal information. Furthermore, differential privacy has a

strong composition property that makes analysis construction from basic components possible. Something that is crucial for building tools and frameworks.

The epsilon parameter (ϵ) is a value to quantify the privacy loss or harm. The privacy parameter has no unit but enables us to compare the effect and risk of two analyses on individuals' privacy. Privacy is a fuzzy concept and different people put different values to their personal data. To understand the privacy risk better, consider the following scenario inspired by [35]. Assume results from a medical survey affects an insurance company's decision on the health insurance coverage. A change in the insurance coverage from participation of other people in the study is unavoidable; however, we expect that the cost (as a factor of risk) that Alice pays for participation in the study does not exceed a certain limit. Differential privacy can help us set these limits, pay compensation and reward people for participation in surveys.

Mechanism Design Randomized response is a well-known method proposed by Warner [48], later modified by Greenberg et al. [22], to give data subjects privacy in the form of deniability. As pointed out by Dwork [8], this mechanism provides differential privacy for answers given by participants in a survey. Suppose, for simplicity, that a yes-no question involves an embarrassing answer. In this method, to respect people's privacy, the interviewer instructs people to flip two coins secretly before responding to yes/no questions. Each question should be answered "yes" if both coins come up head, answered "no" if both coins come up tail. Otherwise the question should be answered truthfully. Keeping the result of the coin toss secret, only each person knows whether the answer reflects the reality or the random coin flip. The respondent can deny this answer by claiming that the coin flip was the reason for the response.

The analyst, knowing that each individual did this procedure before responding to each question can conclude that roughly half of the questions are truthfully answered. From the other half, roughly half of them ($1/4$ of the total responses) are "Yes" and $1/4$ are "No" as a result of the coin flip. The ratio of answers determines a rough estimate of the overall answers. One can further show that this procedure provides $\ln(3)$ differential privacy.

Several more randomisation mechanisms are suggested such as laplacian mechanism [10], exponential mechanism [30], geometric mechanism [21] and stair case mechanism [20]. The laplacian noise provides real values, for integer responses geometric mechanism introduced by Ghosh et al. [21] and for queries that ask for a non-numerical value, like "what is most common disease among database participants?", the exponential mechanism, developed by McSherry and Talwar [30] are applicable.

To use Laplace noise to make a noisy version of function f over a dataset x, one only needs to know the sensitivity of the function, the

maximum change in the result of the function when an item is added to the dataset.

The noisy version of function $f(x)$ that provides ϵ-differential privacy is constructed by adding laplacian noise to the result of the function as below:

$$f'(x) = f(x) + \text{Laplace}(\text{Sensitivity}(f)/\epsilon)$$

If f' is a randomised function constructed from the function f with sensitivity of 1, Fig. 2 shows the probability density function in two neighbouring datasets D and D' for different values of x.

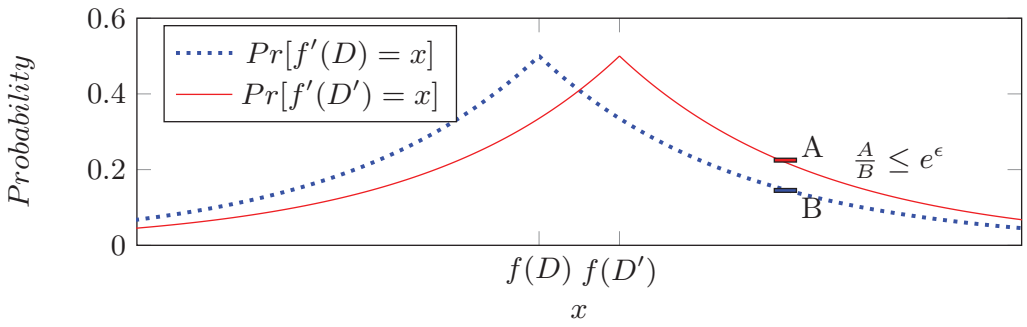

Fig. 2. Probability Density Function

Composition Principle (Sequential) Differential privacy has useful composition properties that make it possible to construct more complex analyses using basic ones. If a family of queries Q_i each give ϵ_i differential privacy, executing a sequence of the queries on the database gives $\sum_i \epsilon_i$-differential privacy. This property, named *sequential composition* is the key for building tools for doing differentially private analyses using basic primitives. In the next section we introduce some of the well-known tools in this area.

3 Variants of Differential Privacy

Apart from the standard and classic definition of differential privacy (Definition 1), different variants of differential privacy for different settings and purposes are proposed. In this section some of these variants are explained.

Centralised and Local Model Trust plays an important role in the perception of privacy. In the centralised model the users have to transmit their data to the curator. Users have to trust the curator for the safe storage of the data and correct usage of differential privacy. The centralised model is not desirable for many users and therefore the local model is seen as an important alternative. While the local model seems to be an

obvious alternative, as shown by many studies, for the same level of privacy these algorithms do not perform as accurately as the centralised model.

Bounded and Unbounded Two variants of differential privacy are commonly used in the field. Bounded differential privacy [27] deals with the case in which neighbouring databases are constructed by changing the value of one individual record in contrast with the unbounded definition with two databases, one with an extra record. This restricts the neighbouring databases to databases with the same size whereas the unbounded differential privacy can be defined for databases with a variety of sizes.

Weakening In (ϵ, δ)-differential privacy [9] (also known as approximate differential privacy), ϵ, as before, represents the effect one individual can have on the information release and δ bounds the probability of a complete privacy breach [41]. As a result, we make room for utility by allowing privacy failure in low-probable events. The case $\delta = 0$ (also known as pure differential privacy) is equivalent of classic differential privacy as explained before.

Generalising In 1970, Alan Westin conducted over 30 surveys about the privacy concerns in different areas. His report demonstrated that people are categorized into three groups: *Privacy Fundamentalists*, *Privacy Pragmatists* and *Privacy Unconcerned* that motivates the need for non-uniform privacy requirements for different individuals and items in a user profile.

Recently, in three independent works [17, 26, 1], some efforts have been made to make differential privacy user centric. We discuss our paper comprehensively in section 6.2 and here we only discuss the two subsequent papers.

Having in mind that people have different privacy preferences, Jorgensen et al. [26] proposed a personalised privacy method that can be specified at the user level. They show that the proposed personalised differential privacy framework has composition properties and a procedure is provided to convert mechanisms from classic differential privacy to their method. To summarize, the data is first sampled non-uniformly and then the right differentially private mechanism is applied. Individual tuples can have different probabilities of being sampled depending on their privacy preferences.

Even for different parts of an individual data record, there may be different privacy preferences and expectations. This non-uniformity of privacy preference brought the idea of heterogeneous differential privacy [1]. In their generalisation of differential privacy, users describe their privacy expectations with a privacy vector. The privacy vector value for each item ranges from zero for the absolute privacy to one for the stan-

dard classic privacy. Later they describe the *Stretching Mechanism* that changes the sensitivity of functions after applying the privacy vector.

Andrés et al. [2] introduce the notion of a *geo-indistinguishability guarantee*. Rather than completely hiding the user's location, this notion of privacy tries to hide locations with some level of approximation within a radius r. While hamming distance is commonly used as a metric of difference between two databases, to adopt distance to the location privacy setting Euclidean distance between locations is used. In addition, a mechanism based on laplacian noise, that let users obtain needed service even after randomised mechanism, is provided.

Different Context Assumptions : Some other research focuses on the usage of privacy in different setting and special contexts. *Pan Privacy* [12] aims to retain the privacy of individuals in a data stream while data is processed, with the assumption of an adversary that can observe the internal state of the system at any unpredictable time. Dwork et al. [11] consider a system in which the result of the same query changes because of a change in the underlying database. This is a common situation for monitoring systems in which the system continuously produces output (this may not be the case for Pan-Private systems). In this setting neighbouring (adjacent) inputs should be defined differently. In *event level privacy*, two databases are neighbours when they differ in one of the events that belongs to a user, whereas in *user level privacy* the databases may differ in multiple events all belonging to one single user.

4 Scientific Methods and Goals

In the first few sections we looked into importance of privacy, naive deanonymisation techniques and scandals caused by them. In further sections we continued by introducing differential privacy as prominent notion and studied its variants. We introduced few privacy enhancing tools and regulatory requirements aiming to improve the current state of user privacy. Among these tools we are particularly interested in programming language approaches that facilitate analysis construction as a program and enforcing the differential privacy dynamically at runtime. We improve these systems by 1) proposing a new privacy accounting schema that permits users to announce their expectation of privacy 2) add new functionalities and operations that are not necessary deterministic. 3) Shifting the trust from the analyst, who performs the analyses, to data owners.

To study and verify the correctness of these systems, we build minimalistic but formal models. These models allow us to abstract away from details of implementation and to focus on fundamental components, their behaviours individually and lastly their interactions with the rest of the system.

5 Differential Privacy, Tools and Methods

Data holders, eager to release information, are not usually privacy experts. Going through different formulas and computing the privacy cost of an analysis is a quantified real-world problem. In addition, an analysis can be arbitrarily complex. A data mining algorithm usually consists of sub-analyses, few iterations and branches on an intermediate result. This brings up the idea of using programming language techniques to ease the construction of these analyses. As a result, many differential privacy frameworks, methods and tools with variety of purposes are introduced. In this section we look at some of them.

Privacy Integrated Queries PINQ [31] is a generic framework to construct differentially private analyses for analysts that are unfamiliar with differential privacy. All programs which are written in this framework and use its APIs, automatically satisfy differential privacy. In PINQ, every data source is wrapped in a PINQ (.NET) object. From this point, the data can be only manipulated via a restricted API that only allows aggregated results to be emitted to the analyst. These primitives are divided into two major categories, transformations and aggregation.[3] Input data is usually transformed before being used in an aggregation function. Aggregation functions work on some specific types of input and these transformations shape the input accordingly. Only stable transformations are allowed in this framework. The stability of a transformation T is c if for any datasets A and A' :

$$|T(A) \oplus T(A')| \leq c \times |A \oplus A'|$$

Some transformations are intrinsically stable like mapping a function over records in a dataset, filtering records based on a predicate, and grouping records into limited groups that share a property. In unstable transformations small differences between neighbouring datasets (when a dataset contains an individual's records and when it does not) may lead to unbounded difference in the resulting datasets. To deal with unstable behaviour of operators like *join*, PINQ introduces a restricted variant with *similar* behaviour.

When data is selected and formed for a statistical analysis, an aggregation operation (like Count, Sum, Average, Median, Min, Max) is executed on it. The result of these aggregation operations are allowed to be emitted to the analyst. Aggregations in PINQ use the laplacian mechanism to add the needed amount of noise. Composition of a c-stable transformation with an ϵ-differential privacy query results in a computation with higher sensitivity that gives $(c \times \epsilon)$-differential privacy. *Function sensitivity* is a numerical value that measures the maximum order of mag-

[3] PINQ has a modular structure and can be extended with new primitives (transformations or aggregations).

nification in distance as a result of applying the function to similar inputs. Low sensitive functions maps nearby inputs to the nearby outputs.

As mentioned before, differential privacy is compositional and queries can always be composed sequentially. In order to improve utilisation and reduce the privacy cost of analyses, *parallel queries* are introduced. Parallel queries are executed on disjoint subsets of records and provide a better lower bound on the privacy guarantee. Assume any arbitrary n queries, $Q_i, (1 \leq i \leq n)$ that each provide ϵ_i-differential privacy in isolation. If these queries are executed on n disjoint subsets of data, the lower bound for total privacy cost decreases from $\sum_{1 \leq i \leq n} \epsilon_i$ to $\max(\epsilon_1 \dots \epsilon_n)$. Parallel queries are especially useful for building histograms in which data points are grouped based on one of the attributes.

wPINQ [37] is a generalisation of PINQ to weighted datasets, capable of answering a larger class of queries like analyses on graphs. Since the amount of noise is significant, considering the fact that the input dataset is not usually the worst case, wPINQ assigns real valued weights to individuals and scales down the contribution of problematic individuals. This avoids high noise magnitude that is added as the result of applying high sensitive functions.

Airavat [39] is the integration of Mandatory Access Control (MAC) to differential privacy in order to provide confidentiality, integrity and privacy on large scale distributed cloud computing. Airavat prevents information leakage by using SELinux-like mandatory access control on the file system, modifying Java virtual machine and enforce differential privacy on its Map-Reduce framework.

Fuzz, DFuzz, Adaptive Fuzz [23, 38, 50] presents a typed functional programming language with a type system equipped with extra metrics to track function sensitivity to find the right amount of noise needed to guarantee differential privacy. Dfuzz [38] introduces dependent types as an extension to Fuzz, allowing a larger class of analyses; among them are analyses that need runtime information.

Fuzz also introduces some practical attacks using time and state side channels. Some precautions are introduced in PINQ to examine and rewrite methods that leak information via side channels. Despite their efforts, Fuzz exhibited their ineffectiveness. An analysis that observes an embarrassing record can run sub queries to use up the entire privacy budget. The exception caused by lack of budget can reveal the existence of the target record. The type system in this functional language infers the privacy cost *statically* which closes the privacy budget side channel. More importantly they introduced predictable transactions (micro queries) that process a single record in the fixed amount of time. They proved that the timing channel can be ruled out by aborting the micro query and replacing the result with a default value provided by the user.

Finally, they proposed network communication to separate the analyser and the machine running the query in order to eliminate electromagnetic radiation, power consumption and cache base side channels.

Relational Algebra [SQL] Relational algebra is commonly used as a theoretical foundation of query languages (like SQL) used to query relational databases. Palamidessi and Stronati [36] analysed and assigned tight sensitivity bounds to relational algebra operators and introduced composition rules that determine how sensitivity of a query can be computed from the sensitivity of the relational operator that has constructed it.

FLEX [25] is a practical tool for enforcing differential privacy for real world SQL queries based on elastic sensitivity. The elastic sensitivity is an upper bound on local sensitivity computed from the database metrics and the query structure.

GUPT [32] Noise magnitude can be adapted by considering the database that the query runs over rather than just the function logic that operates on it. This is achievable, as Nissim et al. [34] stated, by defining smooth sensitivity that represents the variability of a function in the neighbourhood of a particular instance of the database. This is challenging as the noise magnitude may leak information about the content of database and the accuracy of answers depends on the content of dataset. GUPT is a system that demonstrates this instance-based additive noise method. Computing or approximating smooth sensitivity might be difficult. GUPT uses a sample and aggregate framework to tackle this problem. This method can be automated and is applicable for an interactive model in which the function is given as a black-box.

IO Automata Tschantz et al. [47] introduce a formal probabilistic IO automaton model for differential privacy analyses. They use probabilistic bisimulation techniques to verify differential privacy of an interactive system inspired by PINQ. In addition to the proof technique, they show how bounded memory of their system causes increase in the privacy leakage bound.

CertiPriv Barthe et al. [5] introduce the *CertiPriv* framework that assists deriving the differential privacy guarantee provided by a program using *Probabilistic Relational Hoare Logic*. It is built on top of *Coq* proof assistant [46] and is capable of reasoning about approximate differential privacy. The method is powerful enough to build proofs of differential privacy for non-trivial mechanisms. Finally, they illustrate the generality of their approach by providing a correctness proof for several differential privacy mechanisms and algorithms.

RAPPOR [19] One real world tool that uses differential privacy is *Google Chrome* that anonymises users' browser preferences before sending reports to Google servers. Building on top of randomized responses,

RAPPOR sends software reports that look like random data, but can eventually in aggregation, be used to extract useful information. The generated report is only usable to gain statistical information about a user's client software setting without the possibility of inferring the exact user preferences.

Using primitive principles of differential privacy, it is easy to see how anonymisation of one single bit of response is applicable for anonymising bit strings. Taking the advantages of this, numerical or ordinal values can be represented by predicates on different ranges. A novel method of this system is the use of bloom filters to randomize non-categorical domains, like an arbitrary set of strings in a differentially private manner.

PROCHLO [6] is the implementation of the ESA (Encode, Shuffle, Analyze) architecture. The encoding is performed locally to transform, fragment and apply random noise to users' data. The shuffler strips metadata such as IP addresses, timestamps and further uses sampling and reordering of items to de-associate users from reports. Eventually the analyser decodes, aggregates and finally publishes the result.

DPCOMP [24] is a tool that allows researchers to evaluate different algorithms on a collection of datasets. Different factors that are influencing the performance of these algorithms are varied to measure their impacts on the accuracy and utility of the results.

Apple iOS 10 Apple introduced usage of differential privacy in their 10th edition of their operating system (iOS 10) in 2017 [3]. Several patents [44, 45] are registered explaining the details of the algorithms.

6 Contributions

This thesis comprises five papers, "*Featherweight PINQ*" [15], "*Differential Privacy, now it is getting personal*" [17], "*Sampling and Partitioning for Differential Privacy*" [18], "*PreTPost: A Transparent, User Verifiable, Local Differential Privacy Framework*" [16], "*Design and Use of PreTPost Framework*" [13].

Paper I, "*Featherweight PINQ*" published in the Journal of Privacy and Confidentiality, tries to fill the gap between the basic idea of differential privacy and PINQ, the standard and generic framework that implements differential privacy.

Many differential privacy frameworks share the same principles as PINQ. Looking closely at PINQ, one can see some of its deviations from the standard definition of differential privacy. We take a look at the concept of *privacy budget* and how PINQ *dynamically enforces* it, *parallel queries*, *strategies* and the program reflections upon query response. We present Featherweight PINQ, a simplified formal model of PINQ, that explains how PINQ works and proves correctness of the system.

Paper II, "*Personalised Differential Privacy, now it is getting personal*", is published in POPL[4] 2015. The paper tries to improve the budget book-keeping system that is used to *dynamically enforce* ϵ-differential privacy. In this novel method, rather than specifying a global privacy budget for a database, budget assignment is done on user level, hence personalised. Provenance tracking is used to compute the effect of query execution on an individual's budget. We model provenance tracking used in this method and show how this method reduces the privacy cost of analyses.

Paper III, *Sampling and Partitioning for Differential Privacy*, published in the fourteenth annual conference on "Privacy, Security and Trust 2016", looks into to two important basic blocks for constructing a differentially private analysis and their composition principles with existing blocks by introducing a new concept called probabilistic stability.

Paper IV, "*PreTPost: A Transparent, User Verifiable, Local Differential Privacy Framework*", shifts the perspective from a centralised setting to local differential privacy.

Finally the paper V, "*Design and Use of PreTPost Framework*", dives into practical aspects of using the PreTPost framework and the design decisions that are made to serve the framework's requirements.

In the remainder of this section we provide a more in-depth summary of these papers. Papers 1-3 are previously published and presented here unchanged modulo minor typographic corrections and reformatting to fit the style of the thesis.

6.1 Paper I : Featherweight PINQ

PINQ (Privacy Integrated Queries) [31], is one of the well-known tools for constructing differential privacy analyses. PINQ, like LINQ[5], brings the support for various data sources (e.g. DryadLINQ) and usage of general-purpose programming languages to write programs. These features make PINQ an appealing choice for analysers to construct programs that automatically enforce a certain level of differential privacy. The simplicity and extensibility of PINQ empowers experts to introduce variants of differential private frameworks. Among the frameworks that root in PINQ we can list wPINQ [37] for weighted datasets, Streaming-PINQ [49] for streaming algorithms and our tool ProPer for personalised differential privacy (introduced in Paper II).

Many articles have already discussed the theory, compositionality properties and the design of differentially private operations. However, formal verification of an implemented system, like PINQ, requires combining all these blocks and has not been the subject of any of those

[4] 42nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages

[5] Language-Integrated Query integrates query capabilities into the C# language.

studies. In addition, PINQ varies from differential privacy theories by introducing some extra features, the correctness of which have not been thoroughly investigated. In what follows we introduce the three aspects that PINQ handles but which have not been precisely introduced by the theory.

Partitioning Transformation PINQ introduces *parallel queries* by partitioning an input dataset into disjoint subsets. As a result, a parallel query enjoys the maximum of privacy guarantees of all sub-queries, which is lower than the sum of them. This can be explained as one individual from input dataset cannot be present in two disjoint partitions and a query on one partition does not reveal anything about other individuals that are present in the other partitions. This is not correct as the partitioning may also be done in an intermediate dataset rather than the input dataset. Information deriving from one individual may end up in two partitions, since an intermediate dataset may contain two records deriving from one individual.

Strategy for Sequential Queries A differential private program in this framework may inspect the result from one query and determine the parameters of the next query.

Privacy Budget PINQ dynamically enforces a privacy budget for analyses over a sensitive dataset by blocking any query that tries to over-consume the limited privacy budget. The theory only discusses the measuring of privacy cost, whereas PINQ enforces this privacy budget by throwing an exception when the budget is fully exhausted. Observing such behaviour may, in principle, be informative and leak information.

In Paper I, "Featherweight PINQ", we start by introducing the basic foundation of differential privacy that PINQ is based upon, such as ϵ-differentially aggregation operations, stable transformations and their composition with these aggregation operations to construct a query. Then we see how multiple queries are composed (sequential and parallel) and how it affects the overall privacy guarantee that the system promises. Finally, we analyse the gap between the theory and PINQ introduced as a result of:

- dynamically enforcing a certain level of privacy rather than measuring it
- inadequate modelling of adeptness of sequential and parallel queries
- execution of parallel queries on an intermediate table constructed as a result of table transformation

We present a model that simplifies the system in one important aspect, queries that set up a parallel query should be all present in the system on query execution time. To prove differential privacy of Featherweight PINQ, we introduce a probabilistic trace semantic. Using the

model and the trace semantics, we prove that any client program constructed in Featherweight PINQ yields a system satisfying differential privacy. Finally, we suggest two additional safe APIs that allow programs to read the global privacy budget value and the scaling factors for protected tables.

6.2 Paper II : Differential Privacy, Now it is Getting Personal

An important challenge is to improve analyses to gain more information without endangering an individual's information. However, much of the research up to now has tended to focus on improving mechanisms or specific algorithms rather than monitoring what parts of a database are leaked. If an analyst asks about the average salary of men, no information about women in the database is leaked.

Paper II tries to resolves these issues by introducing a new budget book-keeping method. In this setting each individual has a personalised privacy budget. Unlike PINQ in which the sensitivity of a function determines the multiplier of budget reduction from the global privacy budget, in the personalised setting the number of records derived from an individual participating in an analysis determines the multiplier of reduction from this personalised privacy budget. To keep track of records derived from one individual record we use a notion similar to *why provenance* in the terminology of [7]. The significance of the method can be highlighted in the following three points.

Efficient Budget Consumption Parallel queries are introduced in PINQ to reduce the cost of queries on global budget, but constructing parallel queries is cumbersome, because it requires presence of all queries on different partitions at the query execution time. Without the presence of all queries, finding a set of queries that optimises budget cost is not easy. In many cases the immediate answer is needed or there is some dependency between queries.

Good utility is one benefit that is gained from our personalised budget book-keeping technique. Unlike the classic model of budgeting, in which parallel queries play an important role for gaining proper utilization, this method does not rely on parallel queries since the system inherently enjoys efficient budget consumption. This approach also avoids many of the problems (mentioned in the previous section) in modelling parallel queries and designing parallel version of analyses. Assigning ϵ-personalized privacy budget to all individuals guarantees ϵ-differential privacy in the classic model.

Stream Data An added benefit of a personalised budget is in dynamic databases. In a dynamic setting, records may be added or removed from a working dataset continuously. Assigning one global privacy budget, that is decreased as queries are executed, implies that updating the entire database with fresh records does not affect the budget. This is unfortu-

nate since nothing has been leaked about these recent records and one may expect a non-zero privacy budget.

If a privacy budget is assigned to an individual's record when it is added to the dataset and the record can actively participate in analyses until its budget is exhausted, this model of budgeting seems to perfectly fit the requirements of stream databases. We show that our system is capable of answering continuous queries when the data-stream is refreshing the database at a certain rate.

Limitation and Disadvantages Behaviour of the system in corner cases in which particular records participate in more analyses is a controversial case. PINQ blocks a query at the execution time when the global privacy budget is exhausted and this behaviour of PINQ is shown not to leak information. In contrast, the same behaviour in the personalised setting shows the presence of individuals and is considered extremely informative. As an example, if the system blocks a query that intentionally targeted one individual, this clearly indicates that the individual is present in the database. Therefore, this method excludes records that are derived from individuals who do not have a sufficient budget to participate in analyses. Although this makes reasoning about utilization difficult, an analyser with some background information can choose and arrange queries in such a way that error from the absence of depleted records has an insignificant effect on the result.

We present our system, named *ProPer* (Provenance for Personalised Differential Privacy) that is modelled using a similar trace semantics as explained in Paper II with provenance tracking to find track records that are derived from individuals in the input dataset.

Provenance tracking adds overhead to transformation and query execution but comparing execution time of some analyses written in both PINQ and ProPer we demonstrate that the runtime overhead of provenance tracking is modest.

6.3 Paper III : Sampling and Partitioning for Differential Privacy

This paper studies sampling and partitioning methods and their effect on differential privacy in the unbounded differential privacy. We demonstrate a practical attack on PINQ's deterministic sampling transformation. The attack gives an example of when sampling has a negative effect on the differential privacy cost of an analysis compared to the case without sampling.

Furthermore, we investigated other sampling and partitioning methods, especially those that use randomness in their selection and compare them with their deterministic alternatives. To compare and use our result in differential private frameworks such as PINQ, we introduce the concept of probabilistic stability that generalises stability.

We made the interesting observation that only Bernoulli sampling boosts privacy and the partitioning method based on Bernoulli sampling has the same effect as the deterministic parallel composition.

6.4 Paper IV : PreTPost: A Transparent, User Verifiable, Local Differential Privacy Framework

Trust is an important factor in the privacy discussions. In all the previous papers, a trusted party possesses all the data and an analyst or a protection mechanism is responsible to ensure the correct anonymisation of the results. This model, commonly referred to as the centralised model, additionally requires confidentiality for data transformation and security for data storage from both internal and external attackers while the data is stored in the database. Many people do not trust the data curator or see the risk of failure as not acceptable.

The recent adaptation of a local model in the Chrome web browser [19] and the iOS operation system [3] seems to be a step forward toward respecting users' privacy but lack of transparency puts all the efforts in the question. Tang et al. [43] show that the implementation is promiscuous in the choice of the privacy parameter epsilon. Analyses in these systems are either hard-coded in the system or should be communicated to the users. Hard-coded analysis reduces the flexibility while dynamic analysis may endanger the security and confidentiality of the user data.

The analysis that is sent and executed on user data may introduce new security and privacy concerns and makes the need for transparency in the design more severe. These shortcomings bring the idea of a framework with dynamic analysis with transparent design in which the users verify the differential privacy of an analysis themselves with no need to trust the curator.

Pre-processing of the private data in the centralised model requires a complex sensitivity analysis of the algorithm. The post-processing that is common between the local and the centralised setting with the pre-processing theorems in the local model are simple but important fundamental concepts that the PreTPost framework is built upon.

The decomposition of analyses into *Pre, T* and *Post* components and restricting their communication simplifies their isolation and helps the system to block the leakage of information from timing and other types of side channels.

The usefulness of the system is demonstrated by integrating existing local differential privacy algorithms in the PreTPost framework. Additionally, one specific algorithm that is a core to several other algorithms is studied in detail to explain the decomposition technique.

6.5 Paper V : Design and Use of PreTPost Framework

Last section is a tutorial paper that discussed the use and the design of the PreTPost framework. The PreTPost tries to push back the trust

from the analyst who performs analyses to the transparent system that execute the analyses. Therefore, knowing the structure, architecture and the decisions that are made during the design and implementation of the PreTPost framework is important.

Local model tries to minimize the data storage in the curator side therefore the steady communication between users and the curator is important. We see how users can encode their privacy preference as a policy and how the curator and user agree on utility and privacy.

Additionally, we show how the framework can be deployed in a real scenario. We use a scenario of a Internet Service Provider (ISP) who is interested to learn about its client by querying their home routers. In these examples we demonstrate the decomposition of an analysis into pre-processing, randomised transformation and post-processing and present code samples for constructing a query and transmitting it to users.

Finally, we investigate threats from a malicious curator who is trying to extract user's information through unconventional methods. The pre/post processing functions can be a binary executable that may potentially be malicious. We look into the required isolation and sandboxing of different components of the system and how the system overcomes the information leakage through side channels (e.g. timing channel).

6.6 Personal Contributions

"*Featherweight PINQ*" [15] is the simple model we provided for PINQ. The proof of differential privacy for Featherweight PINQ was done by me and the development of the idea and system modelling was jointly done with David Sands.

During my master thesis project, I came up with the idea of a personalised budget and implemented a tool named *PINQuin* [14] to demonstrate it. ProPer (the implementation of which has some minor differences with the original idea of PINQuin) is a powerful model for a system that does provenance tracking in order to dynamically determine who and how much to subtract from the personalised budget. Details of this method are articulated in *Differential Privacy, Now it is Getting Personal* [17]. Modelling and the proof of ProPer were jointly done with David Sands with the help and contribution of Gerardo Schneider.

The idea for the article "*Sampling and Partitioning for Differential Privacy*" [18] came up when I noticed a problem in the PINQ's `Take()` and `Skip()` methods. The proof was developed by me, corrected and improved by David Sands and Thibaud Antignac.

The paper "*PreTPost: A Transparent, User Verifiable, Local Differential Privacy Framework*" [16] describes the framework that is developed after similarities between local differential privacy algorithms are observed and it was decided to unify them in one single framework. The algorithm's integration into the framework by their decomposition into

Pre,T,Post components and the idea of pre-processing theorem were done by me.

Finally I contributed to at least 50% of writing task for each paper except the paper "*Design and Use of PreTPost Framework*" [13] that is fully done by myself.

References

[1] Mohammad Alaggan, Sébastien Gambs, and Anne-Marie Kermarrec. Heterogeneous differential privacy. In *Theory and practice of differential privacy (TPDP 2015) at London, UK*, 2015.

[2] Miguel E. Andrés, Nicolás E. Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, CCS '13. ACM, 2013.

[3] Apple Press Release. Apple previews ios 10, the biggest ios release ever. 2016.

[4] Michael Barbaro and Tom Zeller. A face is exposed for aol searcher no. 4417749. Technical report, The New York Times, 2006.

[5] Gilles Barthe, Boris Kópf, Federico Olmedo, and Santiago Zanella-Béguelin. Probabilistic relational reasoning for differential privacy. *ACM Trans. Program. Lang. Syst.*, 2013.

[6] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Usharree Kode, Julien Tinnés, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. *CoRR*, abs/1710.00901, 2017.

[7] James Cheney, Laura Chiticariu, and Wang-Chiew Tan. Provenance in databases: Why, how, and where. *Found. Trends databases*, 1(4): 379–474, April 2009.

[8] Cynthia Dwork. A firm foundation for private data analysis. *Communications of the ACM*, 54(1):86–95, 2011.

[9] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Proceedings of the 24th Annual International Conference on The Theory and Applications of Cryptographic Techniques*, EUROCRYPT'06, pages 486–503. Springer-Verlag, 2006.

[10] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, TCC'06, 2006.

[11] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In *Proceedings of the Forty-second ACM Symposium on Theory of Computing*, STOC '10, pages 715–724, New York, NY, USA, 2010. ACM.

[12] Cynthia Dwork, Moni Naor, Toniann Pitassi, Guy N Rothblum, and Sergey Yekhanin. Pan-private streaming algorithms. In *ICS*, pages 66–80, 2010.

[13] Hamid Ebadi. Design and use of pretpost framework. 2018.

[14] Hamid Ebadi. PINQuin, a framework for differentially private analysis. Master's thesis, Chalmers University of Technology, 2013.

[15] Hamid Ebadi and David Sands. Featherweight PINQ. *arXiv preprint arXiv:1505.02642*, 2012.

[16] Hamid Ebadi and David Sands. Pretpost: A transparent, user verifiable, local differential privacy framework. 2018.

[17] Hamid Ebadi, David Sands, and Gerardo Schneider. Differential privacy: Now it's getting personal. In *Proceedings of the 42Nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '15. ACM, 2015.

[18] Hamid Ebadi, Thibaud Antignac, and David Sands. Sampling and partitioning for differential privacy. In *14th Annual Conference on Privacy, Security and Trust*. IEEE, 2016.

[19] Úlfar Erlingsson, Aleksandra Korolova, and Vasyl Pihur. RAPPOR: randomized aggregatable privacy-preserving ordinal response. *CoRR*, abs/1407.6981, 2014.

[20] Quan Geng and P. Viswanath. The optimal mechanism in differential privacy. In *Information Theory (ISIT), 2014 IEEE International Symposium on*, pages 2371–2375, June 2014.

[21] Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms. *CoRR*, abs/0811.2841, 2008.

[22] Bernard G. Greenberg, Abdel-Latif A. Abul-Ela, Walt R. Simmons, and Daniel G. Horvitz. The unrelated question randomized response model: Theoretical framework. *Journal of the American Statistical Association*, 64(326):pp. 520–539, 1969.

[23] Andreas Haeberlen, Benjamin C. Pierce, and Arjun Narayan. Differential privacy under fire. In *Proceedings of the 20th USENIX Security Symposium*, 2011.

[24] Michael Hay, Ashwin Machanavajjhala, Gerome Miklau, Yan Chen, Dan Zhang, and George Bissias. Exploring privacy-accuracy tradeoffs using dpcomp. In *Proceedings of the 2016 International Conference on Management of Data*, SIGMOD '16, pages 2101–2104, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3531-7.

[25] Noah M. Johnson, Joseph P. Near, and Dawn Xiaodong Song. Practical differential privacy for SQL queries using elastic sensitivity. *CoRR*, abs/1706.09479, 2017.

[26] Zach Jorgensen, Ting Yu, and Graham Cormode. Conservative or liberal? personalized differential privacy. In *International Conference on Data Engineering (ICDE)*, 2015.

[27] Daniel Kifer and Ashwin Machanavajjhala. No free lunch in data privacy. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, SIGMOD '11, pages 193–204, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0661-4.

[28] Michal Kosinski, David Stillwell, and Thore Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 110(15): 5802–5805, 2013.

[29] Michal Kosinski, Yoram Bachrach, Pushmeet Kohli, David Stillwell, and Thore Graepel. Manifestations of user personality in website choice and behaviour on online social networks. *Machine Learning*, 95(3):357–380, Jun 2014. ISSN 1573-0565.

[30] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, pages 94–103. IEEE Computer Society, 2007.

[31] Frank D McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 19–30. ACM, 2009.

[32] Prashanth Mohan, Abhradeep Thakurta, Elaine Shi, Dawn Song, and David Culler. Gupt: privacy preserving data analysis made easy. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 349–360. ACM, 2012.

[33] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, SP '08, pages 111–125, Washington, DC, USA, 2008. IEEE Computer Society. ISBN 978-0-7695-3168-7.

[34] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 75–84, New York, NY, USA, 2007. ACM.

[35] Kobbi Nissim, Thomas Steinke, Alexandra Wood, Micah Altman, Aaron Bembenek, Mark Bun, Marco Gaboardi, David O'Brien, and Salil Vadhan. Differential privacy: A primer for a non-technical audience (preliminary version), 2017.

[36] Catuscia Palamidessi and Marco Stronati. Differential privacy for relational algebra: Improving the sensitivity bounds via constraint systems. In *Proceedings 10th Workshop on Quantitative Aspects of Programming Languages and Systems*, 2012.

[37] Davide Proserpio, Sharon Goldberg, and Frank McSherry. A workflow for differentially-private graph synthesis. *CoRR*, abs/1203.3453, 2012.

[38] Jason Reed and Benjamin C. Pierce. Distance makes the types grow stronger: A calculus for differential privacy. In *Proceedings of the*

15th ACM SIGPLAN International Conference on Functional Programming, ICFP '10, pages 157–168, 2010.

[39] Indrajit Roy, Srinath T. V. Setty, Ann Kilzer, Vitaly Shmatikov, and Emmett Witchel. Airavat: Security and privacy for mapreduce. In *NSDI*, pages 297–312. USENIX Association, 2010.

[40] Rebecca Skloot. The immortal life of henrietta lacks, the sequel. Technical report, The New York Times, 2013.

[41] Thomas Steinke and Jonathan Ullman. Between pure and approximate differential privacy. *CoRR*, abs/1501.06095, 2015.

[42] Latanya Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, 2002.

[43] Jun Tang, Aleksandra Korolova, Xiaolong Bai, Xueqiang Wang, and XiaoFeng Wang. Privacy loss in apple's implementation of differential privacy on macos 10.12. *CoRR*, abs/1709.02753, 2017.

[44] A.G. Thakurta, A.H. Vyrros, U.S. Vaishampayan, G. Kapoor, J. Freudiger, V.R. Sridhar, and D. Davidson. Learning new words, 2017. US Patent 9,594,741.

[45] A.G. Thakurta, A.H. Vyrros, U.S. Vaishampayan, G. Kapoor, J. Freudinger, V.V. Prakash, A. Legendre, and S. Duplinsky. Emoji frequency detection and deep link frequency, July 11 2017. US Patent 9,705,908.

[46] The Coq Development Team. The coq proof assistant reference manual, 2015. http://coq.inria.fr.

[47] Michael Carl Tschantz, Dilsun Kaynar, and Anupam Datta. Formal verification of differential privacy for interactive systems (extended abstract). *Electron. Notes Theor. Comput. Sci.*, 276:61–79, September 2011.

[48] Stanley L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):pp. 63–69, 1965.

[49] Lucas Waye. Privacy integrated data stream queries. In *Proceedings of the 5th annual conference on Systems, programming, and applications: software for humanity*. ACM, 2014.

[50] Daniel Winograd-Cort, Andreas Haeberlen, Aaron Roth, and Benjamin C. Pierce. A framework for adaptive differential privacy. *Proc. ACM Program. Lang.*, 1(ICFP):10:1–10:29, August 2017. ISSN 2475-1421.