Thesis for the Degree of Doctor of Philosophy

Language-Based Differential Privacy with Accuracy Estimations and Sensitivity Analyses

Elisabet Lobo-Vesga

Department of Computer Science and Engineering
Chalmers University of Technology
Gothenburg, Sweden, 2023

Language-Based Differential Privacy with Accuracy Estimations and Sensitivity Analyses
Elisabet Lobo-Vesga

© Elisabet Lobo-Vesga, 2023

ISBN 978-91-7905-811-1
Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr 5277
ISSN 0346-718X

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

Printed at Reproservice, Chalmers University of Technology
Gothenburg, Sweden, 2023

Language-Based Differential Privacy with Accuracy Estimations and Sensitivity Analyses
Elisabet Lobo-Vesga
Department of Computer Science & Engineering
Chalmers University of Technology

Abstract

This thesis focuses on the development of programming frameworks to enforce, by construction, desirable properties of software systems. Particularly, we are interested in enforcing differential privacy–a mathematical notion of data privacy–while statically reasoning about the accuracy of computations, along with deriving the sensitivity of arbitrary functions to further strengthen the expressiveness of these systems. To this end, we first introduce DPella, a programming framework for differentially-private queries that allows reasoning about the privacy and accuracy of data analyses. DPella provides a novel component that statically tracks the accuracy of different queries. This component leverages taint analysis to infer statistical independence of the different noises that were added to ensure the privacy of the overall computation. As a result, DPella allows analysts to implement privacy-preserving queries and adjust the privacy parameters to meet accuracy targets, or vice-versa.

In the context of differentially-private systems, the sensitivity of a function determines the amount of noise needed to achieve a desired level of privacy. However, establishing the sensitivity of arbitrary functions is non-trivial. Consequently, systems such as DPella provided a limited set of functions—whose sensitivity is known—to apply over sensitive data; thus hindering the expressiveness of the language. To overcome this limitation we propose a new approach to derive proofs of sensitivity in programming languages with support for polymorphism. Our approach enriches base types with information about the metric relation between values and applies parametricity to derive proof of a function's sensitivity. These ideas are formalized in a sound calculus and implemented as a Haskell library called Spar, enabling programmers to prove the sensitivity of their functions through type-checking alone.

Overall, this thesis contributes to the development of expressive programming frameworks for data analysis with privacy and accuracy guarantees. The proposed approaches are feasible and effective, as demonstrated through the implementation of DPella and Spar.

List of publications

Appended publications

This dissertation is for the degree of Doctor of Philosophy, and includes reprints of the following papers:

Paper A	"A Programming Language for Data Privacy with Accuracy Estimations" **Elisabet Lobo-Vesga**, Alejandro Russo, and Marco Gaboardi *ACM Transactions on Programming Languages and Systems (TOPLAS), 2021.* *Observations.* This paper expands upon the results of our published work: "A Programming Framework for Differential Privacy with Accuracy Concentration Bounds" **Elisabet Lobo-Vesga**, Alejandro Russo, and Marco Gaboardi *Proceedings of the 2020 IEEE Symposium on Security and Privacy (SP), 2020.*
Paper B	"Sensitivity by Parametricity: Simple Sensitivity Proofs for Differential Privacy" **Elisabet Lobo-Vesga**, Alejandro Russo, and Marco Gaboardi *Manuscript under submission.*

Acknowledgments

I would like to take this opportunity to express my sincere gratitude to those who contributed and accompany me throughout my doctoral journey. First, I would like to express my deepest gratitude to Alejandro Russo, my supervisor. His patient guidance, inspiring enthusiasm, and constant encouragement have been instrumental in shaping my research and have played a critical role in my academic and personal growth. He has consistently challenged me to think critically and pushed me to improve my skills, including my Street Fighter II abilities – I remain determined to win a match someday! I am truly grateful for his mentorship and advice throughout this whole process. I would also like to express my appreciation to my co-supervisor and collaborator, Marco Gaboardi, for his detailed lessons on probability and type theory. His expertise and guidance have been invaluable, I am grateful for the opportunity to have worked with him.

To my friends and colleagues, I extend my sincere thanks for their fellowship and insightful conversations. Their encouragement has been a source of inspiration and motivation for me. I am forever grateful for the lifelong friendships that we have formed while sharing this experience; it truly has been an honor.

I owe a debt of gratitude to my family who has been my constant support system. Their love, prayers, and encouragement have been a driving force behind my academic endeavors. I am especially grateful to my parents, who instilled in me a love of learning and have supported me every step of the way; and to my brother, the person whom I admire the most and who has paved the way for my success. Finally, words cannot express how thankful I am for my partner Alejandro Gómez; for always believing in me, thanks.

To each and every person who has been part of this experience, I am forever grateful. Your input has made this journey possible, and I am thankful for your presence in my life.

March 1st, 2023

Contents

Abstract iii

List of publications v

Acknowledgments vii

I Introduction 1
- I.1 Privacy protection in context 2
- I.2 Differential privacy 5
  - I.2.1 Properties 6
  - I.2.2 Models and tools 6
  - I.2.3 Challenges 7
- I.3 Statement of contribution 8
  - I.3.1 Addressing challenge 1 8
  - I.3.2 Addressing challenge 2 9

Accuracy

A A Programming Language for Data Privacy with Accuracy Estimations 15
- A.1 Introduction 15
- A.2 Background 19
- A.3 DPella by example 21
  - A.3.1 Basic aggregations 21
  - A.3.2 Cumulative Distribution Function 24
- A.4 Privacy 28
  - A.4.1 Components of the API 28
  - A.4.2 Transformations 30
  - A.4.3 Partition 30
  - A.4.4 Aggregations 32
  - A.4.5 Privacy budget and execution of queries 32
  - A.4.6 Implementation 33
- A.5 Accuracy 33
  - A.5.1 Accuracy calculations 33
  - A.5.2 Implementation 37
  - A.5.3 Accuracy of Gaussian mechanism 40
- A.6 Case studies 42
  - A.6.1 DPella expressiveness 43
  - A.6.2 Privacy and accuracy trade-off analysis 46
  - A.6.3 K-way marginal queries on synthetic data 48
- A.7 Testing accuracy 50
- A.8 API generalization 53

- A.8.1 Implementation and Accuracy estimations 54
- A.9 Limitations & Extensions 56
- A.10 Related work 59
- A.11 Conclusions 61
- Bibliography 63

Sensitivity

B Sensitivity by Parametricity: Simple Sensitivity Proofs for Differential Privacy 69

- B.1 Introduction 69
  - B.1.1 Motivating examples 71
- B.2 $\lambda_{\textsc{Spar}}$: a calculus for distance tracking 73
  - B.2.1 Syntax 73
  - B.2.2 Operational semantics 77
- B.3 Formal guarantees 77
  - B.3.1 Sensitivity by Parametricity 80
- B.4 $\lambda_{\textsc{Spar}}$ as a library 82
  - B.4.1 Vectors 83
  - B.4.2 Currying 85
  - B.4.3 Sorting 86
  - B.4.4 Beyond sensitivity 86
- B.5 Implementation 87
  - B.5.1 Equality for type-level natural numbers 87
  - B.5.2 Spar as an embedded DSL 89
  - B.5.3 Executing functions 90
- B.6 Discussion 91
- B.7 Related work 92
- B.8 Conclusions 94

Appendix 95

- B.A Spar's complete syntax 95
- B.B Typing system 96
- B.C Semantics 99
  - B.C.1 Operational 99
  - B.C.2 Distance and length interpretation 100
- B.D Logical relations 100
  - B.D.1 Definitions 101
- B.E Proof of metric preservation and accompanying lemmas 102
  - B.E.1 Fundamental lemma of logical relations 112
- Bibliography 145

1

Introduction

Constantly sharing personal information has been integrated into our daily routines. From our home devices, online interactions, and the services we use; to more explicit disclosures such as filling out forms and answering surveys, our data is being collected, stored, sold, and processed by a wide range of agents. These agents (e.g., research institutions, government agencies, and businesses) rely on collected data to improve their services, understand populations, tailor policies, and make informed decisions. Consequently, data processing is at the backbone of our society and has the potential to impact our communities and lives positively. It is then desirable to share our information with such agents for personal and societal gains. However, the information provided often contains confidential and sensitive details about ourselves that we expect to remain private and accessible only to those trusted parties; unfortunately, this has not always been the case.

The mishandling of sensitive data has become commonplace among companies and public institutions [17, 14, 21, 27]. As a result, many laws, regulations, and agreements [6, 4, 16, 2] have been put in place recognizing the importance of protecting individuals' privacy and mitigating the occurrence of privacy breaches. Improper disclosure of the information is severely penalized with fines which might put some companies out of business or heavily affect their reputation and competitiveness [28, 1, 19, 3]. To make matters worst, when privacy breaches occur, they are irreversible and have lingering consequences on those affected. These incidents perpetuate distrust between the individuals and the agents interested in their data, deterring the public from sharing their information in the future [10, 7, 11]. The vast implications of privacy breaches then severely limit the potential usage of individuals' data and its availability altogether.

It is in everyone's interest to avoid privacy breaches, but ensuring data privacy is a complex problem. Companies, researchers, and policymakers have searched for robust and concrete ways to define, ensure, and regulate data privacy. Decades of trial and error have made it evident that data privacy cannot be achieved with a few hacks or as an afterthought. Instead, it must be a fundamental approach that can withstand technological changes and unforeseen risks while being feasible for today's needs.

Are our requirements for data privacy utopian? Should data analysis be halted or reduced to preserve individuals' privacy? Fortunately, that is not the case; various privacy-preserving techniques are available that allow us to perform statistical data analyses while guaranteeing the privacy of individual participants. One such approach is differential privacy [13], a mathematical and quantifiable definition of privacy that has gained popularity for its provable guarantees and applicability. Nevertheless, as the problem of privacy is broad and intricate, it is essential to clarify the domain in which differential privacy is applicable and the drawbacks that it might have. Concretely, this dissertation explores some challenges concerning the deployment and usability of differential privacy and addresses them in the context of programming languages.

Before diving into the opportunities and challenges of differential privacy, it is important to explore the context of privacy protection and its threats. Following the reader can find a brief description of some well-known and relevant techniques used by data analysts and privacy practitioners in their daily tasks. This primer will serve as an introduction to the field of statistical data privacy and as a motivation for the usage of the study and application of differential privacy.

I.1 Privacy protection in context

Data anonymization or de-identification. Is the process of removing personally identifying information (PII) from datasets so that the remaining information cannot be linked to specific individuals. In practice, these techniques require data owners to pre-process datasets by purging *explicit identifiability* information such as names and government-issued IDs; as well as *potentially identifiability* information such as IP addresses or next of kin. The remaining data presents a best-of-both-words scenario in which unscrupulous actors will not be able to identify the people providing the information, and honest analysts will have useful data to perform their studies.

The promise of yielding *useful and privacy-preserving* results has positioned anonymization techniques as the *de-facto* approach among practitioners storing, sharing, and processing sensitive data. This sense of assurance is further reinforced by regulatory agents and globally common statutes in which anonymization is considered *sufficient* to protect individuals' privacy [26]. Despise its apparent robustness, data breaches still occur in the presence of anonymized data.

The weakness of anonymization techniques is their incapacity to account for data's multiple degrees of identifiability. While PIIs are indeed attributes an adversary can use to identify an individual, the same result can be achieved by combining attributes that do not classify as personally identifiable. For instance, Sweeney [31] demonstrated that the combination of ZIP code, birth date, and sex are unique to 87% of the American population. Furthermore, when considering other available sources of information, the probability of uniquely identifying individuals is increased by cross-referencing with the anonymized data. It is then clear that data anonymization is susceptible to privacy attacks and cannot always fulfill its promise of providing useful and privacy-preserving results.

Privacy attacks on anonymized data aim to reverse the process of anonymization. Attackers can exploit the aforementioned vulnerabilities by associating anonymized records with non-anonymized information from different datasets, this technique is known as a linkage attack. Using non-anonymous data as background knowledge, attackers are capable to trace back individuals (known as re-identification attacks) or recover large portions of the original dataset (known as reconstruction attacks). Even though these attacks might seem difficult to perform and unlikely to succeed, concrete instances of such attacks abound. Consequently, I present two infamous cases in which sophisticated data administrators overestimated anonymization guarantees and compromised the privacy of hundreds of people.

- **AOL Searcher No. 4417749**: To provide useful data for academic research, AOL released a dataset of search queries performed by its users. The company anonymized said data by replacing user IDs with random numbers and removing IP addresses. The combination of searches performed by a user —whose identity was hidden behind an associated random number— were naively considered non-identifiable attributes of that user. Later on, New York Times journalists Barbaro and Zeller, prove this assumption to be false [8]. In the article, the authors showcase how a set of searches can reveal particular characteristics of the users. Concretely, they re-identified and presented user No. 4417749, a 62-year-old widow searching for "numb fingers", "60 single men", "dog that urinates on everything", "homes sold in shadow lake subdivision gwinnett county georgia.", and "landscapers in Lilburn, Ga,". When notified about the vulnerabilities, AOL removed the dataset and apologized for its publication, but, as pointed out by the authors, the data was already copied and distributed on other sites; thus leaving AOL users' permanently exposed.
- **Netflix competition**: Netflix released a dataset containing movie ratings provided by their users as part of training data for a competition to improve their recommendation algorithm. To anonymize the dataset, user IDs were replaced, several ratings were randomly altered and dates were modified. Despise their efforts, Narayanan and Shmatikov [25] demonstrated that more 80% of the users were identifiable by knowing the time and rating of only three movies. By using publicly available ratings from the Internet Movie Database (IMDB) as background knowledge, the authors were able to re-identify common users across the datasets, in addition to learning other potentially sensitive information such as users' apparent political preferences.

These examples exhibit the prevalence of using anonymization for privacy preservation among practitioners, but more importantly, they demonstrate the theoretical and practical limitations of this technique, casting substantial doubts about anonymization's power for ensuring privacy.

Summary statistics. A common refrain among data analysts is to "aggregate" data to make it safe to share and release. The idea behind this approach is that it provides a hide-in-the-bunch effect where individuals are not likely to be singled out. Intuitively, this simple approach fulfills the promise of protecting individuals' privacy, after all, how can an attacker know my specific salary if all that is shared is the average income of people in my area? As it turns out, this intuition is full of risks and potential mistakes.

Region	Age	Sex	Count
A	20 - 29	F	-
A	30 - 39	F	20
A	40 - 49	F	9
A	50 - 59	F	17
A	20 - 59	F	49

Table I.1: Summary female population in region A

Straight-forward attacks can be foreseen under the presence of outliers or when the population is not big enough to "hide" data points. In fact, the field of statistical disclosure control [29] aroused from the need to protect information on tabular and aggregated data. Consequently, statistical organizations have devised various methods to mitigate these attacks, among them, the *threshold rule* stands as the most commonly used [5]. The threshold rule consists on requiring a minimum number of respondents (per categorization) in order to provide the aggregated results. For instance, applying a threshold of 5 would mean that at least 5 individuals must share the same combination of age, sex, and region of residence in order to provide any insights about a population with this categorization. Although the threshold rule is simple to implement and seemly efficient to prevent issues with identifying eccentric data points; the privacy guarantees are broken when the aggregated statistics are reversible and the releases are accumulated through time.

Consider the aggregated data in Table I.1 containing the summary statistics of the female population in a certain region. Here the population of females is aggregated within age ranges, additionally, the total population of females (known as a marginal statistic) is provided. With this information, we can easily identify that number of females between the ages of 20-29 is $49 - 20 - 9 - 17 = 3$. Even though this example presents an obvious scenario, reversing aggregations across many dimensions when marginal statistics are included is a well-known and common problem [9].

Releasing marginal statistics jeopardizes the privacy guarantees provided by summary statistics, however, privacy-by-aggregation's vulnerabilities exist beyond marginal summaries. When aggregated data is produced over time, attackers are provided with additional information that can be used to compare and infer sensitive information. Say our previous example was produced for January, in which {**Region**:A, **Age**:40-49, **Sex**: F} $= 9$; if the results of the following month are {**Region**:A, **Age**:40-49, **Sex**: F} $= 10$, it is easy to notice that one person has been added thus, violating the threshold rule since less than 5 individuals are represented in the difference between both counts. To make matters worst, if an attacker has previous knowledge of Alice moving to region A in this period, they can infer that Alice is between 40 to 49 years old.

The main problem with privacy-via-aggregation is that all methods of numerical aggregation can be used to reconstruct the original data. This phenomenon was called the *Fundamental Law of Information Recovery* by Dwork and Roth [13] stating that "overly accurate answers to too many questions will destroy privacy in a spectacular way". Intuitively, the more statistics generated from a single set of data, the greater the chance of reconstructing the original data from those statistics; this is simply because each release decreases the possibilities for the data that could have produced those statistics. This is why prominent data managers including the U.S. Census Bureau [24], Google [15], and Apple [18], have shifted their interest to more robust tools for releasing privacy-preserving statistics such as differential privacy.

I.2 Differential privacy

Differential privacy [13] is a formal mathematical definition of privacy in aggregate statistics (e.g., averages and histograms) and machine learning analysis (e.g., k-means and stochastic gradient descent). This formal framework has gained increasing popularity during the past decades as its core mechanisms are a variant of the classic *randomized response* [32], protecting individuals' privacy with formal guarantees of *plausible deniability*—i.e., when performing a statistical analysis over a dataset, any participant can deny the presence of their information in the input data. Accordingly, differential privacy ensures that anyone observing the result of a differentially-private computation will likely make the same inferences about an individual, whether or not their information is included as input for the analysis. Furthermore, differential privacy specifies mathematical assurance for privacy protection against various privacy attacks such as re-identification, reconstruction, and differencing attacks.

The success of differential privacy lies in the fact that it identifies algorithms as the primary culprits for data breaches. Under differential privacy, data is not anonymized, as we have seen that this technique is susceptible to linkage attacks [30, 25, 8, 12]. Additionally, differential privacy does not rely on the potential privacy of aggregated statistical results, as this approach is susceptible to reconstruction and membership attacks [20, 12]. Instead, differential privacy focuses on how the algorithms at hand can influence the relationship between the input (possibly sensitive) data and the outcome of the computations. In this sense, differential privacy is not a single tool or implementation but a *criterion* or *property* that many algorithms for accessing sensitive personal data are devised to satisfy.

Intuitively, an algorithm (often referred to as query or analysis) is said to satisfy differential privacy when it returns statically indistinguishable outputs when given two datasets differing in the data of a single individual. In order to fulfill this condition, differentially-private algorithms add calibrated noise to their result to mask the absence, inclusion, or modification of someone's information in the input dataset. The strength of differential privacy's guarantees can be tuned via the *privacy parameter* ϵ[1]. This parameter is commonly referred to as the *privacy loss* as it can be interpreted as the additional risk a participant is exposed to by partaking in a specific data analysis. Consequently, the value of ϵ directly influences the noise in a computation's result to ensure privacy. As ϵ decreases, the strongest the privacy guarantees are; however, this comes at the cost of adding more noise, thus impacting the results' accuracy.

---

[1] In its general form, differential privacy is parametrized by (ϵ, δ), with ϵ bounding the total privacy loss and δ referring to a failure probability of the DP guarantees.

At the core of every differentially-private algorithm lies the noise calibration mechanism. Noise calibration is crucial to provide both useful and private results. While the ϵ parameter quantifies the desired level of privacy, we also need to consider how susceptible the algorithm is to disclose an individual's information when the dataset changes. The quantification of how much an operation's result changes relative to its inputs is known as *sensitivity*. Together, ϵ and the algorithm's sensitivity provide us with enough information to determine how much noise is needed to achieve differential privacy.

I.2.1 Properties

The rigorous mathematical guarantees provided by differential privacy yield several practical benefits for its users [13]:

Composability. Differential privacy features beneficial compositional properties allowing analysts to create complex analyses using basic ones. The principle of *sequential composition* is one of the most basic ones stating that if a family of algorithms $\mathcal{A}_i$ satisfy ϵ_i differential privacy, then executing a sequence of the algorithms satisfies $\sum_i \epsilon_i$-differential privacy. The principle of *advanced composition* can produce tighter limits on their total privacy loss when considering iterative algorithms.

Provable guarantees. Differential privacy is the only existing approach providing provable privacy guarantees for successive data releases.

Transparency. Differentially private algorithms and their parameters are not secrets to be protected. Opposite to traditional de-identification tools, knowing the extent to which data has been transformed does not threaten the differential privacy guarantees. This transparency boosts reproducibility, accountability, and public trust in the process of data analysis.

Post-processing resilience. Differential privacy guarantees that any subsequent processing of data releases does not increase the risk of privacy violation for individuals.

Group privacy. While differential privacy is commonly used to protect privacy at the individual level, it has been shown that its guarantees also translate to (weaker) protection for groups of individuals. Concretely, an algorithm satisfying ϵ-differential privacy for individuals also provides $k\epsilon$-differential privacy for groups of size k.

I.2.2 Models and tools

Since differential privacy is a mathematical property, there are multiple ways in which we can design algorithms to fulfill it. Depending on whether data collectors are trusted, differentially private algorithms can be executed centrally or locally. In a centralized setting, the individuals transmit their raw data to trusted parties; it is assumed that these entities will safely store the data and correctly use differential privacy to access the information. In contrast, in a local setting, data collectors are not trusted; therefore, each participant will perturb their response before sharing; hence sensitive information is never stored in one location.

The local model seemingly provides the ideal scenario where privacy is guaranteed, and security is boosted by avoiding creating honeypots for hackers. However, several studies have shown that these algorithms do not perform as accurately as those in the centralized model for the same level of privacy. Consequently, most differential privacy tools—including those introduced in this dissertation—are based on the centralized model.

Most frameworks for differential privacy are based on the same principle: they provide a set of fundamental private analyses, which the analysts can use as building blocks to create more complex algorithms. This approach relies heavily on the compositional property as this principle will determine the final privacy guarantees of the combined analyses.

I.2.3 Challenges

Privacy-accuracy reasoning

Composability is a fundamental property for developing programming tools for differential privacy. When combining building blocks, these tools ensure that the total privacy loss of the resulting analysis does not exceed the desired privacy level. This characteristic facilitates reasoning about an analysis' total privacy loss as a budget that is distributed and spent through the algorithms' pieces.

Strongly connected to privacy is the concept of accuracy. Analysts might be interested in controlling their algorithms' privacy and accuracy. One could argue that privacy is a concern solely for the individuals (and data holders), while accuracy is a concern exclusively for the analysts (and those interested in the statistical analyses). Unfortunately, reasoning about accuracy is less compositional than reasoning about privacy. Determining the accuracy of arbitrary user-defined algorithms is complex as it depends on the specific task at hand and the specific error measurement. In the literature, most of the standard algorithms for differentially-private analyses are provided with accuracy estimations (in the form of confidence intervals or error bounds); however, the accuracy of their combination is addressed on a case-by-case basis. As a result, most programming frameworks for differential privacy do not offer any support for tracking, reasoning, and adjusting the accuracy of the algorithms; the crucial task of predicting accuracy is left to the analysts.

Proof of sensitivity for user-defined functions

Noise calibration is at the backbone of every differentially private algorithm. To sample the adequate noise required to satisfy differential privacy, we need to consider the desired privacy level (ϵ) and the sensitivity of the algorithm at hand (a measurement of how volatile it is to changes in its inputs). Unfortunately, determining the sensitivity of arbitrary operations can be challenging. For this reason, most programming tools for differential privacy do not support the definition of arbitrary operations. Instead, they are equipped with predefined operations whose sensitivity is known, avoiding sensitivity calculations altogether. However, even though predefined operations have allowed for many exciting analyses, it severely constrains the kind of computations we can perform on the datasets, thus limiting access to valuable information.

Several programming frameworks have been proposed to compute the sensitivity of user-defined operations. The typical approach to statically computing the sensitivity of a program consists of providing a language with a type system enriched with sensitivity annotations. Then, when combining the provided primitives, the type system will keep track of the program's global sensitivity. Unfortunately, most of these frameworks are never fully deployed because they often rely on advanced features not available in mainstream programming languages, thus requiring creating full-stack languages from scratch. Moreover, those frameworks that manage to create a functioning prototype lack acceptance by data analysts since the tools are based on niche programming devices (e.g., linear and modal types) unknown outside academic circles.

I.3 Statement of contribution

This dissertation encompasses a series of works proposing several programming techniques to help non-experts write differentially private algorithms and reason about the different components of these algorithms. With the deployment of such techniques, we expect to equip data analysts with tools where i) they can create data analysis satisfying differential privacy by construction, ii) they can reason about the privacy-accuracy trade-offs before execution and, iii) they are not limited to a set of predefined algorithms to create their own.

At a high level, the contributions of this dissertation can be grouped into two categories, each of them tackling one of the challenges listed above:

I.3.1 Addressing challenge 1

We created DPella, a programming framework for differentially-private algorithms that allows data analysts to reason compositionally about privacy-accuracy trade-offs at compile time. DPella's main novelty is that it exemplifies how programming frameworks can internalize the use of probabilistic bounds for composing different confidence intervals or error bounds in an automated way. DPella leverages taint analysis to detect statistical independence of the noise added by its different primitives; this information is then used to achieve better error estimates. Finally, since DPella's analysis is data-independent, it showcases how mainstream statically-typed languages can be used to perform differential privacy analysis as part of their type-checking process without relying on any runtime execution or information.

These results are recorded in our work "A Programming Language for Data Privacy with Accuracy Estimations." *In 2021 ACM Transactions on Programming Lan-*

guages and Systems (TOPLAS) [23] which in turn is an extension of our previous work "A Programming Framework for Differential Privacy with Accuracy Concentration Bounds." *In 2020 IEEE Symposium on Security and Privacy (SP)* [22]. The former is the only one included in this dissertation as it encompasses both results; the main differences between both works are highlighted in a subsequent section A.1.

I.3.2 Addressing challenge 2

We proposed a sound calculus (λ_{SPAR}) for statically determining the sensitivity of user-defined programs while avoiding using linear and relational refinement types. Our approach relies on a novel use of parametricity—a well-known abstract uniformity property enjoyed by polymorphic functions—together with type constraints and type-level naturals to verify a program's sensitivity by simply type-checking. Its simplicity facilities embedding λ_{SPAR} into mainstream richly-typed programming languages.

We introduced SPAR, a concrete implementation of λ_{SPAR} as a library for the Haskell programming language. The library SPARis implemented as an embedded domain-specific language which allows us to leverage Haskell's advanced type inference to provide some support for sensitivity inference via type error—a feature that, to our knowledge, has not been explored before. Finally, we complemented our findings with the implementation of classic examples (such as summing, mapping, and sorting elements of a vector) to demonstrate how SPARcan be used to prove user-defined programs' sensitivity. The main result of this work opens the door to integrating procedures for automatically proving the sensitivity of user-defined analyses into the programming workflow, e.g., by using SPAR's sensitivity proofs as an input to other Haskell-based DP frameworks.

Bibliography

[1] IBM: Cost of a data breach report. *Network Security*, 2022(8):4, 2022.

[2] AB-375. California consumer privacy act (CCPA). 34, 2018.

[3] A. Acquisto, A. Friedman, and R. Telang. Is there a cost to privacy breaches? an event study. In *ICIS 2006 Proceedings - Twenty Seventh International Conference on Information Systems*, pages 1563–1580, 2006.

[4] P. Act. Family educational rights and privacy act (FERPA). 2014.

[5] L. Arbukle. Aggregated data provides a false sense of security. `https://iapp.org/news/a/aggregated-data-provides-a-false-sense-of-security/`, April 2020. [Online; posted 27-April-2020].

[6] U. G. Assembly et al. Universal declaration of human rights. *UN General Assembly*, 302(2):14–25, 1948.

[7] G. Bansal, F. Zahedi, and D. Gefen. The impact of personal dispositions on information sensitivity, privacy concern and trust in disclosing health information online. *Decision Support Systems*, 49(2):138–150, 2010.

[8] M. Barbaro and T. Zeller Jr. A face is exposed for AOL searcher no. 4417749. `https://www.nytimes.com/2006/08/09/technology/09aol.html`, August 2006. [Online; posted 09-August-2006].

[9] L. Buzzigoli and A. Giusti. From marginals to array structure with the shuttle algorithm. *Journal of Symbolic Data Analysis*, 4(1):1–14, 2006.

[10] H. Choi, J. Park, and Y. Jung. The role of privacy fatigue in online privacy behavior. *Computers in Human Behavior*, 81:42–51, 2018.

[11] M. J. Culnan and P. K. Armstrong. Information privacy concerns, procedural fairness, and impersonal trust: An empirical investigation. *Organization Science*, 10(1):104–115, 1999.

[12] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 202–210, 2003.

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

[14] H. Elizabeth A. and N. Perlroth. For target, the breach numbers grow. *The New York Times (January 10)*, 2014.

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

[16] European Commission. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), 2016.

[17] J. Finkle and D. Skariachan. Target cyber breach hits 40 million payment cards at holiday peak. *Reuters (December 18)*, 2013.

[18] A. Greenberg. Apple's 'differential privacy' is about collecting your data—but not *Your* data. `https://www.wired.com/2016/06/apples-differential-privacy-collecting-data/`, June 2016. [Online; posted 13-June-2016].

[19] M. Henriquez. $4.35 million – the average cost of a data breach. *Security*, 59(10):7, 2022.

[20] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J. V. Pearson, D. A. Stephan, S. F. Nelson, and D. W. Craig. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS genetics*, 4(8):e1000167, 2008.

[21] N. Horton and A. DeSimone. Sony's nightmare before christmas: The 2014 north korean cyber attack on sony and lessons for us government actions in cyberspace. Technical report, JHUAPL Laurel United States, 2018.

[22] E. Lobo-Vesga, A. Russo, and M. Gaboardi. A programming framework for differential privacy with accuracy concentration bounds. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 411–428. IEEE, 2020.

[23] E. Lobo-Vesga, A. Russo, and M. Gaboardi. A programming language for data privacy with accuracy estimations. *ACM Trans. Program. Lang. Syst.*, 43(2), June 2021.

[24] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber. Privacy: From theory to practice on the map. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, pages 277–286.

[25] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125. IEEE, 2008.

[26] P. Ohm. Broken promises of privacy: Responding to the surprising failure of anonymization. *UCLA l. Rev.*, 57:1701, 2009.

[27] S. A. O'Brien. Giant equifax data breach: 143 million people could be affected. *CNN Tech*, 8, 2017.

[28] J. Ruohonen and K. Hjerppe. The gdpr enforcement fines at glance. *CoRR*, abs/2011.00946, 2020.

[29] C. Skinner. Chapter 15 - Statistical Disclosure Control for Survey Data. In C. Rao, editor, *Handbook of Statistics*, volume 29 of *Handbook of Statistics*, pages 381–396. Elsevier, 2009.

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

[31] L. Sweeney. Uniqueness of simple demographics in the us population. *LIDAP-WP4, 2000*, 2000.

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