THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Differential Privacy - A Balancing Act

BOEL NELSON

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2021

Differential Privacy - A Balancing Act

Boel Nelson

ISBN 978-91-7905-487-8
Serial number 4954 in *Doctoral theses at Chalmers University of Technology*.
New series ISSN0346-718X

Technical Report No. 200D
Department of Computer Science and Engineering
Chalmers University of Technology
412 96 Göteborg,
Sweden
Phone: +46 (0)31 772 10 00

Author e-mail: `boeln@chalmers.se`

Printed by Chalmers digitaltryck
Göteborg, Sweden 2021

Differential Privacy - A Balancing Act

Boel Nelson
Department of Computer Science and Engineering
Chalmers University of Technology

ABSTRACT

Data privacy is an ever important aspect of data analyses. Historically, a plethora of privacy techniques have been introduced to protect data, but few have stood the test of time. From investigating the overlap between big data research, and security and privacy research, I have found that *differential privacy* presents itself as a promising defender of data privacy.

Differential privacy is a rigorous, mathematical notion of privacy. Nevertheless, privacy comes at a cost. In order to achieve differential privacy, we need to introduce some form of inaccuracy (i.e. error) to our analyses. Hence, practitioners need to engage in *a balancing act* between accuracy and privacy when adopting differential privacy. As a consequence, understanding this accuracy/privacy trade-off is vital to being able to use differential privacy in real data analyses.

In this thesis, I aim to bridge the gap between differential privacy in theory, and differential privacy in practice. Most notably, I aim to convey a better understanding of the accuracy/privacy trade-off, by 1) implementing tools to tweak accuracy/privacy in a real use case, 2) presenting a methodology for empirically predicting error, and 3) systematizing and analyzing known accuracy improvement techniques for differentially private algorithms. Additionally, I also put differential privacy into context by investigating how it can be applied in the automotive domain. Using the automotive domain as an example, I introduce the main challenges that constitutes the balancing act, and provide advice for moving forward.

Keywords: accuracy, accuracy/privacy trade-off, big data, data privacy, differential privacy, privacy, utility, vehicular data

Preface

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

- **Boel Nelson**, Tomas Olovsson, "Security and Privacy for Big Data: A Systematic Literature Review", in *2016 IEEE International Conference on Big Data (Big Data)*, Washington, DC, 2016, pp. 3693-3702, doi: 10.1109/BigData.2016.7841037.
- **Boel Nelson**, Tomas Olovsson, "Introducing Differential Privacy to the Automotive Domain: Opportunities and Challenges", in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, Toronto, ON, 2017, pp. 1-7, doi: 10.1109/VTCFall.2017.8288389.
- Mathias Johanson, Jonas Jalminger, Emmanuel Frécon, **Boel Nelson**, Tomas Olovsson, Mats Gjertz, "Joint Subjective and Objective Data Capture and Analytics for Automotive Applications", in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, Toronto, ON, 2017, pp. 1-5, doi: 10.1109/VTCFall.2017.8288366.
- **Boel Nelson**, "Randori: Local Differential Privacy for All", preprint
- **Boel Nelson**, 2021. "Efficient Error Prediction for Differentially Private Algorithms". To appear in *The 16th International Conference on Availability, Reliability and Security (ARES 2021)*, August 17–20, 2021, Vienna, Austria. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3465481.3465746.
- **Boel Nelson**, Jenni Reuben, "SoK: Chasing Accuracy and Privacy, and Catching Both in Differentially Private Histogram Publication", in *Transactions on Data Privacy* 13:3 (2020) 201 - 245.

Acknowledgments

They say it takes a village to raise a child, and the same is probably true when it comes to finishing a PhD. I'd like to say a big "THANK YOU!" to all the people who intentionally or unintentionally have played big parts on this journey.

First off, the senior staff working closest to me. To professor David Sands for always listening and trying to understand me. To associate professor Niklas Broberg who always believed in me, mentored me and taught me to be a better teacher. To associate professor Vincenzo Gulisano who would always understand, and listen to me complaining when my code didn't work. To associate professor Magnus Almgren for planting the seed of curiosity regarding differential privacy all those years ago. To associate professor Tomas Olovsson for seeing my potential and hiring me. To all of my co-authors and collaborators. Without you, this thesis would never have seen the light of day.

Next, I'd like to acknowledge all the strong role models and friends that inspire me. To my amazing Mika who motivates me to keep going when the going gets hard. To Jenni who always inspires me to dream and reach higher. To Hedvig, who has shown me that you can achieve anything as long as you put your mind to it. To Bei whom I admire both for her kind spirit and awesome research. To Rashidah who remains playful and curious through every challenge, even though she doesn't even have access to a washing machine back home...! To

Alexandra who has shown me that it's possible to come out on top of things. To Moyra, who inspires me through her systematic approach and assertive way of expressing herself. To Laleh, my soul sister. To Elin, who's probably the sassiest, coolest and most confident person I know. To all the people who have made an impact that I surely forgot to mention. Your actions inspire me!

To my friends and colleagues at Chalmers. Thank you for all the lunch trains, the chats in the lunch room, and more recently, our Zoom sessions. To Max and Alexander who would always make me laugh... or cry (don't remind me of your number of citations please) depending on the context. To all the past and present PhD students in iSec and NS: Irene, Benjamin, Mattí, Nashi, Elisabet, Agustín, Alejandro, Iulia, Mohammad, Ivan, Carlos, the Daniels, Wissam, Valentin, Thomas, Iosif, Elena, Carlo, Nasser, Thomas, Aljoscha, Joris, Babis and Georgia. To Sandro and Pablo, I'll never understand how both of you can seem so relaxed and cool as post docs, but you give me hope for the future! To Andrei and Alejandro, who both somehow make *professor-ing* look easy. To Wolfgang, who told me all those years ago to "Just do IT!". To Anthony who always inspires me to improve my writing and teaching skills. To Fia, who makes me think we can change the world, one course at the time. All of you make Chalmers a better place!

Not to forget, all the people who keep the department running. Especially a big thank you to Marianne. You keep all of us afloat! To Agneta who helped me when I struggled with my first SLR. To everyone involved in Programrådet that provided an outlet for discussions, and made me believe anything is possible when we work together. To everyone that has contributed to making my journey as smooth as possible, thank you!

To all of those who couldn't or *wouldn't* complete the PhD. I've learned that the price of the journey can be high, and that it's not worth the toll for everyone. There's no shame in choosing yourself over the degree. This thesis is dedicated to all of you.

To Lin-Manuel Miranda, who won't read this, but still inspired me to both *not*

throw away my shot and also *write my way out*. The Hamilton soundtrack is what played all the way through my last paper and the introduction to this thesis. As such, I've also realized that the Hamilton soundtrack can work as a unit to measure time. For example, one round of proof-reading Paper V took about 1 Hamilton, and addressing the comments about 1.3 Hamiltons. Funnily enough, my opponent is situated in Hamilton, NY. Perhaps this is poetic justice at work?

Lastly, I'd like to thank my family and extended family. To Bris and Mika (again!) who always have my back, who listen to all my struggles and small victories. You two will always be my family of choice! To my OG role model, Dr dad. Look at me dad, almost a Dr! To my brother who always thought I'd use my "talent" to make money... I guess that failed. Моим русским "родителям" Наталье и Григорию. To my wonderful Andrej. You are my favorite person!

Boel Nelson

Göteborg, March 2021

Contents

Introduction

BIG data has been a buzz word for years. Everyone wants it, and with the rise of machine learning and artificial intelligence research, there seem to be many use cases for big data as well. Outside of the research community, data is also becoming increasingly important. Not only can data be used as decision support, there are also companies that solely collect and sell data. As such, data is in high demand.

Still, with great data comes great responsibility. First of all, privacy is a universal right, recognized both in the Universal Declaration of Human Rights (UDHR) [1] and the European Convention on Human Rights (ECHR) [2]. And, secondly, there are huge fines[i] [3] involved with breaking data regulations in the EU. Despite the clear societal and legal demand for data privacy, data breaches still occur frequently.

Only over the past few years, data breaches have resulted in the unintentional disclosure of millions of user's private information [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. To make matters worse, data breaches are irrevocable. Once data has been released there is no taking it back. In some cases, the effect of a data breach can be mitigated. For example, suppose my password was leaked — I can easily change the password. Similarly, I can, with some additional hassle, request a

[i] "*up to 10 000 000 EUR, or in the case of an undertaking, up to 2% of the total worldwide annual turnover of the preceding financial year, whichever is higher*"

new credit card if my credit card gets stolen. In other cases, data breaches result in permanent damages. For example, imagine my medical records got stolen, as in the Brazilian case [14]. There is no way that I can change my medical diagnoses just because the information has been leaked. As such, *data leaks are forever*, and data should be protected with this danger in mind.

Now, do we need to give up our dreams about data analysis in order to preserve privacy? What if there was a way to protect individual's privacy, while still getting accurate results? In fact, there exists such a solution, achieved by ensuring *differential privacy*. Now, this thesis would not exist if the solution to all privacy problems was as simple as just applying differential privacy. Of course, differential privacy comes with its setbacks, and this thesis is dedicated to addressing some of those setbacks to bring differential privacy one step closer to being a practical solution for real world data collections.

Additionally, differential privacy is not the first or only proposed solution to achieve data privacy. In fact, there exits many other approaches. Historically, many of these privacy-preserving techniques have been introduced in conjunction with statistical analysis of data in databases. As such, I want start off by introducing the reader to the origins of data privacy in the context of database research.

1.1 A Historical Perspective on Database Privacy

First, let me introduce some terminology that I will use throughout this thesis that relates to databases. When talking about data, I will use the words *database* and *data set* interchangeably. Using terminology from databases, I will refer to one *row* of a database table (Table 1.1) with an individual's data as a *record*, and the *columns* as *parameters* or *attributes*. For example, {€, Boel, Female, PhD student, Information security, Swedish} is a record, and Salary is an attribute. I will also use either the word *query* or *algorithm* to refer to some data analysis, such as calculating an average or a sum. To be able to exem-

plify database queries, I will use my own style of pseudo code that to an extent reassembles the language SQL. Having established some terminology, we can now move on to discuss database privacy.

Salary	Name	Gender	Title	Division	Nationality
€€€	Alice	Female	Professor	Networks and systems	Greek
€€€	Bob	Male	Professor	Information security	British
€	Boel	Female	PhD student	Information security	Swedish
€€	Eve	Female	Associate professor	Networks and systems	Swedish

Table 1.1: Example database with 4 records

Databases have been around for decades, with the CODASYL query language being introduced in late 1960's [15]. Relational databases, that still are popular today, was introduced by Codd [16] in 1970. Already in the early days, both security and privacy was studied in the context of databases. One particularly persistent privacy problem is the *inference problem*. In fact, this thesis revolves around solutions to the inference problem, and their application. So, what is this inference problem, and why is it so hard to solve?

The inference problem states that sensitive data can be inferred through *indirect disclosures*. For example, if an adversary wants to learn my salary, directly running the following query would most likely be disallowed since the name in this example database is a unique identifier:

```
SELECT salary FROM employees WHERE name=Boel
```

In fact, obvious identifiers such as names and social security numbers would most likely already be removed from the database, since they would cause direct disclosures of sensitive information. So what constitutes an indirect dis-

closure? Assume that the adversary knows some *background data* about me, such as what division I belong to and what nationality I have. Now, using the background data, the adversary can ask for:

```
SELECT salary FROM employees WHERE gender=female AND
 title=PhD student AND division=information security
                AND nationality=Swedish
```

While this second query may seem like an innocuous query, this query would in fact only target me, even if the example database contained all staff at my university. That is, the query would match exactly one record in the database. At glance, it may seem easy to just disallow queries that returns one record, but as we will learn, this solution still allows indirect disclosures.

Historically, there have been many attempts to counteract the inference problem. Specifically, within database research, several *inference control* techniques have been proposed. Here, the objective of inference control is to "*make the cost of obtaining confidential information unacceptably high*" [17]. That is, the purpose of data privacy has never been to completely stop the flow of information, but rather to limit the leakage of confidential information. As Dalenius [18] recognized, information leakage (disclosure) can happen when releasing census statistics [19]. Hence, when releasing data, information leakage is expected, but the caveat here is that when guaranteeing privacy, we want to be able to quantify said information leakage.

Initially, the inference problem was addressed by sanitizing before data was released [20]. That is, the database itself was sanitized before it could be queried. Then, in 1970, Hoffman and Miller [21] started investigating the inference problem in online settings, where the database can be queried dynamically. Studying this online setting further complicates the inference problem, as it requires us to also reason about what can be learned when we *compose* results from multiple queries. Imagine that we try to block indirect disclosures by restricting queries that return only one record. How could an adversary still figure out what my salary is? For example, the adversary can first query the database for everyone's

salary:

```
SELECT salary FROM employees
```

Which would return €€€€€€€€€. Then, the adversary runs a query that asks for everyone's *except my salary*:

```
SELECT salary FROM employees WHERE NOT name=Boel
```

Which returns €€€€€€€€. Now, the difference between the two queries is €, which is my salary. As such, an adversary can use set theory to figure out which queries to run in order to disclose information of their choice. Consequently, blocking queries that return only one record does not solve the inference problem. Instead, we would much rather want a method that prevents adversaries from comparing queries that *differ on only one record*.

After some years of new solutions to the inference problems being suggested Denning and Schlörer [20] systematized the existing inference control techniques in 1983. They identified two types of inference controls: those that put restrictions on the allowed queries, and those that add noise to data or query results. Furthermore, they also grouped the inference controls as follows:

- Restriction techniques
  - Table restriction (coarse-grained)
  - Cell restriction (fine-grained)
- Perturbation techniques
  - Record-based perturbation (pre-processing)
  - Result-based perturbation (post-processing)

The restriction techniques disallow access either to whole tables or specific cells. For example, adding the requirement to disallow queries that target a single record can be categorized as a restriction technique. In hindsight, I would like to point out that the restriction techniques has evolved primarily to today's research on *information flow control* and *access control*. As such, in this thesis I will not discuss restriction techniques in relation to data privacy. Instead, variations of the perturbation techniques are what is mainly used to achieve privacy in data analyses.

Most notable in the context of differential privacy, is the introduction of random disturbances to achieve privacy, which Olsson [22] attribute to Statistics Canada. This technique falls within the perturbation techniques. Here, Olsson [22] propose using randomness to choose how numbers should be rounded, and can thus be used either as pre-processing or post-processing. For example, when a query returns the exact value 1.5, we can post-process the answer to instead return 2 with probability $\frac{1}{3}$, and 1.5 with probability $\frac{2}{3}$.

Also interesting in the context of differential privacy, is the idea of *Random Sample Query* control by Denning [23]. Here, the records included in the query are chosen randomly to prevent the adversary from calculating which queries differ by only one record.

So, did Denning and Schlörer [20] find a clear champion among the existing techniques? As usual, the answer was *it depends*. More specifically, Denning and Schlörer [20] conclude their survey with an understanding that rings as true today as when they wrote it:

> *The best strategy for a particular application will depend on its objectives and risks.* — Denning and Schlörer [20]

Given the needs for practitioners to identify the 'best' strategy for their data, it is of course important to be able to compare approaches somehow. One such way is by introducing strategies that comes with privacy metrics. We note that several researchers have proposed metrics to quantify anonymity [24, 25, 26, 27, 28]. Still, these metrics are primarily used in communication networks and not databases. As such, we do not dig further into these specific metrics.

Instead, we focus on another branch of privacy metrics that are directly connected to databases. Here, a series of incremental metrics have appeared, sometimes called syntactic anonymity models [29]. All of these models leave the implementation details to the practitioners. Basically, the syntactic anonymity models capture a property of the database instead of arguing for the use of specific methods. In my opinion, it is interesting to notice that the research at this point has moved away from the bottom-up approach of inventing new perturba-

tion techniques.

First out among the syntactic models was k-anonymity [30]. The idea behind k-anonymity is that if there exists k records that look *similar* it will result in a hide-in-the-group effect which provides privacy for the individuals. To explain k-anonymity, we re-use the small example database from before but remove the names (Table 1.2). Here, the sensitive attribute that we want to protect is {Salary}. The example database contains no unique identifier, but instead has four attributes {gender, title, division, nationality} that together form a *quasi-identifier*. A quasi-identifier is a set of parameter that does not on their own uniquely identify a record, but which together create a unique identifier. For example, {Female, PhD student, Information security, Swedish} is a quasi-identifier for Boel.

Salary	Gender	Title	Division	Nationality
€€€	Female	Professor	Networks and systems	Greek
€€€	Male	Professor	Information security	British
€	Female	PhD student	Information security	Swedish
€€	Female	Associate professor	Networks and systems	Swedish

Table 1.2: Example database with 4 records

Now, for the database to have k-anonymity, we need to make sure there are k records that share the same values for their quasi-identifiers. Basically, we need to create groups with k similar records in each group. Depending on the data in one's database, k-anonymity can be achieved in different ways. Hence, we will have to make decisions on a case-by-case basis when we make a database k-anonymous. From our example database, we can for example start by suppressing the attribute nationality to get Table 1.3.

Salary	**Gender**	**Title**	**Division**	**Nationality**
€€€	Female	Professor	Networks and systems	*<redacted>*
€€€	Male	Professor	Information security	*<redacted>*
€	Female	PhD student	Information security	*<redacted>*
€€	Female	Associate professor	Networks and systems	*<redacted>*

Table 1.3: Example database with 4 records, with nationality suppressed

Next, we need to determine which values we want to preserve. For example, we can take Alice's and Eve's record and put them in the same group by generalizing their titles to 'faculty', resulting in Table 1.4. Alice and Eve now belong to a 2-anonymous group with the quasi-identifier {Female, Faculty, Networks and systems}. That is, we have $k = 2$ because there are two records that share the same values for their quasi-identifier.

Salary	**Gender**	**Title**	**Division**	**Nationality**
€€€	Female	Faculty	Networks and systems	*<redacted>*
€€	Female	Faculty	Networks and systems	*<redacted>*

Table 1.4: Example database reduced to 2 records that are 2-anonymous

Still, k-anonymity is sensitive to *group disclosure*. Suppose Alice and Eve has the same salary. Then we would have Table 1.5, which is also 2-anonymous.

Salary	Gender	Title	Division	Nationality
€€€	Female	Faculty	Networks and systems	<*redacted*>
€€€	Female	Faculty	Networks and systems	<*redacted*>

Table 1.5: Example database that is 2-anonymous, but still leaks the salary of anyone belonging to the group

Having noticed that k-anonymity results in privacy breaches when all members of a group share the same value for the sensitive parameter, ℓ-diversitywas introduced [31]. That is, the group described in Table 1.5 is 2-anonymous, but since everyone has the same salary, we would accidentally leak the salary of all female faculty in network and systems. As a remedy, ℓ-diversity introduced requirements for values to be *well represented* within each group.

Due to similar weaknesses, t-closeness [32] and later β-likeness [33] was introduced. From this line of research, it becomes clear that it is difficult to anticipate and protect against all possible attacks. Even though the syntactic anonymity models do not directly take a bottom-up approach, each improvement assumes a slightly stronger adversary. As such, I would not consider the syntactic anonymity models strictly top-down approaches. So, perhaps we would have better luck if we turned the problem around and started with a top-down approach where we start with the strongest adversary possible?

A privacy definition that does not make any assumptions about the adversary's access to background data is differential privacy [34]. Unlike the previous solutions to the inference problem, the foundations of differential privacy does not rely on the data achieving certain properties. Instead, differential privacy is a property of the algorithm that runs the data analysis. As such, differential privacy is fundamentally different in the way it tackles the inference problem than previous solutions. Accordingly, in my opinion, differential privacy can be thought of as a top-down solution, whereas previous solutions take a bottom-up or a mixed approach.

Still, like the perturbation techniques we introduced before, differential privacy also relies on adding some sort of "noise" or randomness to the result of a data analysis. In the end, Denning and Schlörer [20]'s quote on how to achieve data privacy still summarizes the current position of many researchers, although the exact techniques to produce "noise" have been updated:

> *How do you keep John Doe anonymous when reviewing data on the larger picture? Techniques such as adding "noise" to the data help protect the privacy of the individual.* [20]

1.2 Putting Privacy into Context

So, what cases do we want to protect against when we talk about data privacy? Basically, we are interested in protecting an individual's data from being leaked during data analysis. Next, I will introduce two possible attackers that we would like to defend against. We will continue using the example database from the previous section.

Salary	Name	Gender	Title	Division	Nationality
~~€€€~~	Alice	Female	Professor	Networks and systems	Greek
~~€€€~~	Bob	Male	Professor	Information security	British
€	Boel	Female	PhD student	Information security	Swedish
~~€€~~	Eve	Female	Associate professor	Networks and systems	Swedish

Table 1.6: Example database with 4 records, where the highlighted records are already known to the adversary

First, imagine a worst case scenario with a malicious adversary that wants to infer my salary from a database. We assume a strong adversary with access to arbitrary background data. For example, assume that the adversary has access to everyone's salary except mine, i.e. they know all the highlighted information in Table 1.6. Then, if we allow the adversary to query the database for everyone's salary, they can infer what my salary is. Hence, we want to protect individuals even in cases where the adversary has access to *arbitrary background data*.

Now, such a strong adversary might seem uncommon and unrealistic. Does that mean we do not need to need to protect against the extreme case where all but one piece of data is known? While the assumption of access to arbitrary background data might seem too strong, this assumption means we do not need to be concerned about what other data already exists, or what data may be released in the future. As such, this assumption will allow us to give privacy guarantees that do not depend on data outside of our database.

Next, recall the example from Section 1.1 where the adversary can use set theory to construct several queries that target one single record. Unlike before, let us assume a benign user. This user has no intention of inferring any private information. However, if the database does not enforce adequate privacy, even a benign user can accidentally infer information. Imagine that the user wants to query database for everyone's salary, but the database only holds my salary. Or, imagine that the users runs the innocuous query mentioned previously:

```
SELECT salary FROM employees WHERE gender=female AND
 title=PhD student AND division=information security
              AND nationality=Swedish
```

At my university, this particular query would have one unique match, namely me. Imagine running this query on the small example database in Table 1.7. In both these cases, the benign user would unintentionally learn my salary. As such, there are cases where even a benign user ends up targeting data from a single user.

Salary	Name	Gender	Title	Division	Nationality
€€€	Alice	Female	Professor	Networks and systems	Greek
€€€	Bob	Male	Professor	Information security	British
€	Boel	Female	PhD student	Information security	Swedish
€€	Eve	Female	Associate professor	Networks and systems	Swedish

Table 1.7: Example database with 4 records, where third row is the record accidentally being targeted

Therefore, we would like to protect every record in the database in such a way that even when a single record is targeted, the released information still protects the privacy of the single record. That is, we want a notion of privacy that protects the privacy of each record even when two queries only differ by one record.

1.3 Differential Privacy

Differential privacy [34] is a rigorous privacy definition with statistical privacy guarantees. That is, differential privacy allows us to *quantify* privacy through a privacy loss parameter ε. Unlike the k in k-anonymity, ε is a measure of risk.

In addition, differential privacy is a *property of an algorithm*. That is, in order to ensure differential privacy, we only need to verify that the algorithm is differentially private. The alternative would be a *property of data*, where for every possible output, we would need to check that particular output has the property. For example, k-anonymity, ℓ-diversity, and t-closeness are all examples of property of data.

Now, recall the different adversaries we discussed in Section 1.2. How can we make sure each record in a database gets the same level of privacy? We need to define what it means to *differ by one record*, which we established in Section 1.1 was the main reason behind the inference problem. Hence, we start of by defining neighboring data sets in Definition 1.

Definition 1 (Neighboring Data Sets)**.** *Two data sets, X and X', are neighboring if and only if they differ on at most one element x_j. That is, X' can be constructed from X by adding or removing one single element x_j:*

$$X' = X \pm x_j$$

To protect against a strong adversary, we want the results from two *similar* (neighboring) databases to be *almost* the same. In other words, if we go back to the salary example, a differentially private query for everyone's salary vs everyone except my salary should return *almost* the same value with high probability. That is, the probability distributions for the results should be almost the same [35]. Hence, differentially private versions of these two queries should return *almost* the same value statistically:

1. `SELECT salary FROM employees`
2. `SELECT salary FROM employees WHERE NOT name=Boel`

As such, the results will not let the adversary infer my salary. Now, differential privacy is a definition (Definition 2) that allows us to quantify how big this *almost the same* should be in any given context.

Definition 2 (ε-Differential Privacy)**.** *A randomized algorithm f' gives ε-differential privacy if for all data sets X and X', where X and X' are neighboring, and all $\mathcal{S} \subseteq$ Range(f'),*

$$Pr[f'(X) \in \mathcal{S}] \leq e^{\varepsilon} \times Pr[f'(X') \in \mathcal{S}]$$

In addition, differential privacy is also:

- Resistant against arbitrary background data
- Composable
- Safe against any post-processing

That is, not only does differential privacy protect against the cases I introduced in Section 1.2 by not making assumptions on the adversary's access to background data, differential privacy also has additional benefits. First, differential privacy is composable, which means that a user can combine results from different queries and we can still calculate the privacy guarantees. That is, we can execute queries in an online setting as opposed to relying on a pre-sanitized database. Next, any post-processing of the results from a differentially private algorithm will not degrade the privacy guarantees. Basically, we can treat data released by a differentially private algorithm as non-sensitive data.

1.3.1 Centralized vs Local

Seeing as differential privacy is a definition, there are a multitude of ways to achieve differential privacy. One thing that can be tweaked is where the differentially private algorithm is executed. Next, I will introduce two different *modes* of differential privacy: centralized differential privacy, and local differential privacy (LDP).

Centralized differential privacy is what is used in the original paper [34] on differential privacy. In this mode, all data is stored centrally before the differentially private algorithm is executed, as illustrated in Figure 1.1. For example, when an analyst collects data in a database and then runs a differentially private query on the database, this corresponds to the centralized mode.

In contrast, in the local mode (Figure 1.2) the differentially private algorithm is executed before the data leaves the participant. For example, when a participant enters data into a website, and the client runs the differentially private algorithm before sending the perturbed data to a webserver, this corresponds to the local mode. As a consequence, with the local mode we do not need to store all sensitive data in one location. That is, we can avoid creating potential honeypots for hackers by using the local mode. As such, LDP inherently provides a defense against data breaches.

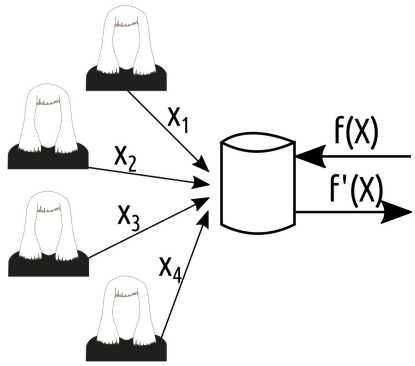


Figure 1.1: Participants raw data (x_n) is gathered in a centralized database, and each query ($f(X)$) on the database is answered under differential privacy ($f'(X)$)

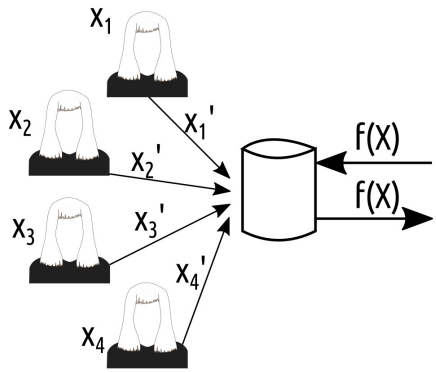


Figure 1.2: Each participant's data (x_n) is perturbed by a differentially private algorithm locally before their data (x'_n) sent to the centralized database

Figure 1.3: The two modes of differential privacy

Basically, choosing between the central or local mode boils down to who can be trusted with the data. That is, the assumption on trust and potential risk of storing data is what decides what mode a data collector should choose. Some

differentially private algorithms are also more suitable for a particular setting. In the next section, I will introduce one algorithm for each mode.

1.3.2 Differentially Private Algorithms

Many algorithms are differentially private, or can be made differentially private by adjusting them to adhere to Definition 2. As such, I cannot introduce all differentially private algorithms. Instead, I will introduce two common differentially private algorithms in this section. First, the *Laplace mechanism*, which is typically used in the centralized mode. And next, *randomized response* which is typically used in the local mode.

The Laplace Mechanism

The Laplace mechanism was one of the first mechanisms introduced by Dwork et al. [34]. One can think of the Laplace mechanism as a result-based perturbation technique, where noise is added to a query result. We illustrate this in Figure 1.4.

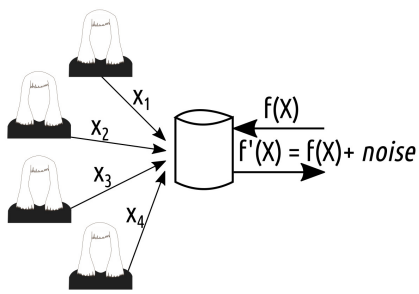


Figure 1.4: The Laplace mechanism adds noise drawn from a Laplace distribution to true result ($f(X)$) to release an approximate result ($f'(X)$)

That is, the added noise is what ensures Definition 2 holds. As a consequence,

for each implementation of the Laplace mechanism, we need to understand how big the noise should be to protect our data. Basically, the noise should make two neighboring data sets X and X' *almost* indistinguishable when we execute the query f' on them. That is, if my salary is the only record that differs between two queries, the noise needs to be big enough to hide that difference. To complicate matters further, my salary needs to be hidden no matter what other data is present in the database. For example, if we add Bill Gates, who presumably would require a very high salary, to our database in the future, our queries need to be able to handle this case too. As such, we need our noise to be big enough to hide the biggest *possible* difference between $f'(X)$ and $f'(X')$. Conveniently enough, there already exists such a measurement, namely the ℓ_1 *sensitivity* (Definition 3 inspired by [36]) of an algorithm.

Definition 3 (ℓ_1 Sensitivity)**.** *Given any algorithm f with numerical output, with any neighboring data sets X, X' as input, the ℓ_1 sensitivity (Δf) of the algorithm f is:*

$$\Delta f = max||f(X) - f(X')||_1$$

Having established how big our noise needs to be, to cover even the worst case difference between two data sets, we can now define the Laplace mechanism as follows in Definition 4.

Definition 4 (Laplace Mechanism)**.** *Given any algorithm f with numerical output and with sensitivity Δf, the Laplace mechanism is defined as:*

$$f'_{Lap} = f + Lap(\Delta f/\varepsilon)$$

Historically, adding noise to data to achieve privacy is a known privacy technique, as we saw from Denning and Schlörer [20]'s 1983 survey. So, how is the noise in the Laplace mechanisms different from those techniques that failed in the past? In particular, how is the randomness in differential privacy different from the technique used by Olsson [22]?

First of all, the noise in the Laplace mechanism is drawn from a known Laplace distribution. As such, we can estimate approximately how big the noise will be.

Secondly, the magnitude of the noise is based on the sensitivity of the algorithm. And lastly, we never add the same noise twice. What difference does fresh noise make? Suppose I first execute the following query:

```
SELECT salary FROM employees WHERE name=Boel
```

The database will then answer with `salary+noise`. Then, I execute a dummy query which returns no one's salary, for example `SELECT salary WHERE name=Boel AND NOT name=Boel`, which just returns `0+noise`. Now, if the noise is the same in both cases, my salary can be easily deduced by subtracting the noise from the first query. Hence, each query needs to get fresh noise.

In conclusion, the Laplace mechanism is conceptually similar to the random noise used historically. Still, the Laplace mechanism is a bit more sophisticated, since we can reason about how queries compose and estimate the accuracy of a query.

Randomized Response

Randomized response [37] is a somewhat odd differentially private algorithm in the sense that it predates the definition of differential privacy by more than 40 years. For that reason, randomized response is an excellent case to exemplify that differential privacy is merely *a property of the algorithm* as opposed to a specific implementation.

In 1965, Warner [37] designed randomized response as a survey answering technique to avoid bias in participants' answers. Basically, when a participant has to answer a sensitive question, they are more likely to lie. For example, a participant may tell us that they never cheated on a test, simply because they are too embarrassed to admit they cheated. Randomized response was designed to counteract this answer bias.

The protocol behind randomized response consists of a few simple steps. First,

the participant is asked a question. Next, the participant decides what to answer in the following way:

1. Flip a coin
a) If **tails**, answer truthfully
b) If **heads**, spin a spinner with all possible answer alternatives to determine what to answer

During randomized response, we assume that the outcome of the coin-flip and spinner is only observed by the participant. Consequently, when I respond that I have cheated on a test, there is no way to tell if that is my true answer or a randomly chosen answer. That is, I can deny that I have cheated on a test, and blame my answer on randomized response. As such, randomized response gives *plausible deniability* to the participant.

Still, as with the Laplace mechanism, the amount of perturbation introduced by randomized response is known and controlled. That is, the bias of the coin and the spinner is known. Hence, we can predict how many of the responses are lies and how many are truthful in order to estimate the true answer distribution. We define randomized response in Definition 5.

Definition 5 (Randomized Response)**.** *A randomized response for any input* x *and some probability* $1 > p > 0$ *is defined as:*

$$f'_{RR} = \begin{cases} x & \textit{with probability } p \\ \neg x & \textit{with probability } 1 - p \end{cases}$$

Having introduced differential privacy, and exemplified with two different algorithms, we can now move on to the heart of this thesis. Namely: the balancing act of the accuracy/privacy trade-off.

1.3.3 Setting ε

The privacy of any differentially private algorithm is expressed as privacy loss through the parameter ε. Accordingly, differential privacy does not make any

claims about data becoming “anonymous”. Instead, ε can be thought of as the additional risk a participant is exposed to by participating in a data analysis.

Consequently, a central part of any differentially private algorithm is setting ε. Deciding on an adequate value of ε is an open and unsolved problem. Moreover, what constitutes a ’good’ value for ε is context dependent [38]. There have been papers written on possible ways of reasoning when setting ε, such as using an economic approach [39], calculating the adversary’s advantage [40], and similarly, calculating the Bayesian posterior belief of the adversary [41]. Generally, there exists a consensus that ε should be set to a small value, which Wood et al. [42] argues should be less than 1. Still, there exists no silver-bullet for setting ε.

Figure 1.5: Abstract view of the trade-off, where the size and placement of the colored areas depend on context and the actor viewing the spectra

The reason setting ε is so difficult is, in my opinion, due to the fact that there are at least two parties with conflicting interests involved. Participants ideally want perfect privacy. Simultaneously, data analysts ideally want perfect accuracy. As such, there will never be a clear cut optimal value to set ε to. Basically, the two parties would need to agree upon which ranges are acceptable (the green area as illustrated in Figure 1.5). Hence, ε needs to be set on a case-by-case basis.

Adding to the complexity of the problem, privacy is a social construct that is difficult to reason about without involving real participants. Since this thesis is written from an engineering perspective, I do not attempt or claim to solve

the problem of giving participants a 'good' value of ε. Instead, I think our best attempt is to tackle the problem from the other end: namely by finding the worst error a data analyst can tolerate. This gives the participants the highest privacy the analysts can afford. Accordingly, we will instead focus on giving participants the lowest ε the data analysts can accept, i.e placing ε to the leftmost in the green area in Figure 1.5.

1.3.4 Accuracy

Strongly connected to privacy is of course accuracy. A critique I have often encountered is that differential privacy is an unrealistically strong notion of privacy. In particular, a concern is that it may not be possible to achieve adequate accuracy under differential privacy. That is, until we can provide upfront accuracy predictions for all algorithms, differential privacy risk not being a strong contender in real data collections. In addition, we would also like to be able to claim that our predicted accuracy is 'good'.

Consequently, my particular focus has been on trying to address the *balancing act* of differential privacy, i.e. the inherent accuracy/privacy trade-off at the heart of differential privacy. As with the privacy loss parameter, ε, accuracy is highly context dependent. Therefore, it is important to balance accuracy and privacy on a case-by-case basis. For example, $\pm$10mg may be negligible when measuring ingredients for your cake mix, but $\pm$10mg could be lethal when measuring medicine. That is, different settings have difference tolerance for inaccuracy. Consequently, our balancing act needs to be put into context before we can reason about 'good' accuracy.

While there exists techniques to improve accuracy of differentially private algorithms without decreasing privacy (see Paper VI), increasing ε is the most obvious way of improving accuracy. Still, there exists other variables, often algorithm dependent, that can affect accuracy such as the size of the database. So, how can we reason about accuracy in this balancing act?

First, we need to find at least one method to measure error in our setting. There are of course different ways of predicting the error of a differentially private algorithm. In this thesis, I have used two fundamentally different methods. The first method, *Chernoff bounds*, is an analytical method that predicts the range the error will fall in. A general Chernoff bound for the accuracy of any differentially private algorithm [43], is given in Definition 6.

Definition 6 ((α, β)-usefulness)**.** *Let f' be a differentially private algorithm, and* E *be a random variable representing the error of the output of f'. Given two error bounds α and β, the population size n and $\alpha, \beta \in (0, \frac{1}{2})$, where $\beta = 2e^{-2\alpha^2 n}$. We say that f' is* (α, β)-useful *[44] if and only if with probability at least $1 - \beta$, the error* E *is bounded above by α, i.e.,*

$$Pr[\mathrm{E} \leq \alpha] \geq 1 - \beta$$

Secondly, I empirically measure error from experiments. For example, I have measured *mean absolute percentage error* (MAPE) empirically. In Paper V, I present a novel methodology for expressing error by creating a prediction model from empirical data. For an example with two parameters A and B, this prediction model y is as follows:

$$y = \gamma_0 + \gamma_1 \times A + \gamma_2 \times B + \gamma_{12} \times AB + \text{experimental error}$$

Where the constant γ_0 is the intercept, and AB is included to capture the possible *interaction* between parameters A and B.

1.4 Thesis Objectives

The original objective of my PhD project was to investigate how privacy can enable big data analysis. Most of the results do not necessarily require 'big' data though, although big data can increase the accuracy of the results in some cases. In particular, I have focused on how to use differential privacy in real data analyses. Motivated by differential privacy's rigorous theoretical guarantees, I set out to encourage the use of differential privacy in real settings. That is, I

want bridge the gap between theory and practice by making differential privacy more accessible to practitioners. I have done this by 1) exploring how differential privacy can be applied in a specific domain and use case, and 2) through practical tools.

Moreover, making differential privacy accessible is not only about bringing tools and techniques to potential users. A major caveat to adopting differential privacy is ensuring adequate accuracy. At the same time, for participants, the main concern is being guaranteed adequate privacy. These two conflicting interests leaves us with a trade-off between accuracy and privacy that needs to be balanced.

In order to bring differential privacy to real systems, adequate levels of accuracy and privacy must be achieved simultaneously. Hence, I focus on differential privacy's inherent privacy-accuracy trade-off in three ways: 1) through tools where data analysts can tweak the trade-off through tangible parameters, 2) by systematizing known techniques for achieving accuracy improvements, and 3) through proposing a novel application of a methodology for creating error prediction models.

In summary, the research questions I address in this thesis are:

- ▷ What privacy model(s) are suitable for big data? (Paper I)
- ▷ How can differential privacy be achieved in the vehicular domain, and are there any additional challenges that apply in this domain compared to existing theoretical research? (Paper II, Paper III)
- ▷ How can we create a tool that gathers poll data under differential privacy by design? (Paper IV)
- ▷ Can we use empirical methods to predict error for differentially private algorithms? (Paper V)
- ▷ How can we improve accuracy of differentially private analyses in other ways than tweaking the privacy parameter ϵ? (Paper VI)

1.5 Summary and Contributions of Included Papers

The papers included in this thesis cover several different topics, all related to data privacy. In Figure 1.6 I have attempted to capture the relationship between all the included papers and their topics. Paper I (page 43) is the most general paper since it only briefly touches on the topic of differential privacy. All the other papers are specifically focused on differential privacy. In Paper II (page 79) we explore how differential privacy can be used in the automotive domain, and what challenges it entails. Paper III (page 103) is a bridge between the automotive domain and tools for differential privacy that Paper IV (page 123) belongs to. In Paper V (page 155), I introduce a methodology for accuracy prediction that may be applicable to any differentially private algorithm. Finally, in Paper VI (page 197) we investigate accuracy for two specific data types: histograms and synthetic data.

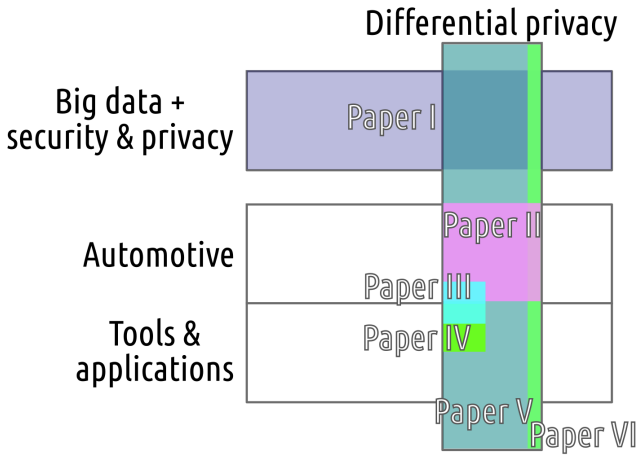


Figure 1.6: Abstract illustration of the relationship between the papers (semi transparent colored areas to show overlaps) and which research areas they cover. The size of areas and papers do not necessarily correspond to their real 'size'.

1.5.1 Paper I

To investigate the intersection between big data research and security and privacy research, we conducted a systematic literature review (SLR) that created a snapshot of the current research field. We found that privacy is currently a popular topic to combine with big data research, and that differential privacy is particularly often used. Our conclusion is that differential privacy is especially well-suited for big data analysis, as it provides mathematically proven privacy guarantees that prevents overfitting of data that would lead to inference of information about individuals. Additionally, as differential privacy is a property of the algorithm and not the data, it is easier to check for than checking properties of the data. Our contribution in this paper is a systematic categorization of recent research papers that span both research areas. We answer the following research questions:

- ▷ What recent security or privacy papers exists in the big data context?
- ▷ How many papers cover security or privacy for big data?
- ▷ Which security, privacy and big data topics are represented in the area?
- ▷ When a paper covers more than one category, which categories intertwine?

Statement of Contribution

My contribution to this paper includes choice of methodology, design of SLR protocol, reading and analysis of papers. In addition, the paper was written by me.

1.5.2 Paper II

In this paper, we connect differential privacy to the automotive domain. Our main goal with this paper was to bridge the gap between theory and practice, by establishing the possible role of differential privacy within the context of the automotive domain, while identifying the challenges of differential privacy bring.

This paper consists of a comprehensive introduction to differential privacy, and focus especially on what challenges can arise when implementing differential privacy in a vehicular setting. Furthermore, we give advice for practitioners as to where to start when implementing differential privacy in this domain. Lastly, we highlight the currently open research problems that apply to the entire differential privacy research community, and also discuss the specific problems encountered when dealing with vehicular data. Thus, the contribution of this paper is as follows:

+ a comprehensible introduction to differential privacy, including what type of differentially private analyses can be performed in the vehicular domain
+ recommendations for how to proceed when implementing differentially private analyses in the vehicular domain
+ highlights of the challenges involved with implementation of differentially private algorithms in the vehicular domain

Statement of Contribution

My contribution to this paper includes comparisons of differentially private algorithms, advice for practitioners based on interactions with project stakeholders, and an analysis of open challenges. In addition, the paper was written by me.

1.5.3 Paper III

Moving on to a particular use case within the automotive domain, we implemented a smartphone app that collects subjective data from drivers. In this paper we showcase how both subjective and objective data can be collected from connected cars simultaneously. The idea is to capture how drivers experience certain scenarios right when it happens, rather than sending a poll in paper format months later. Consequently, the smartphone app collaborates with the in-vehicle network in order to send polls to driver's when interesting scenarios occur. We also discuss what privacy implications our specific use case has

for users, and propose a privacy architecture that relies on differential privacy to guarantee privacy. Our contribution is to provide answers to the following questions:

- ▷ How can we design the subjective data capture app in a way that makes it easy and safe to use in a vehicle, even while driving?
- ▷ How can we design a triggering mechanism to decide when a particular question or set of questions should be posed to a particular user? The triggering mechanism must be versatile and flexible to be usable for all relevant use cases.
- ▷ How can we cater for follow-up questions that depend on answers to previous questions?
- ▷ How can we protect the privacy of users while at the same time providing automotive engineers with as powerful data collection and analytic tools as possible?

Statement of Contribution

My contribution to this paper includes a concept idea for how to add privacy to a system containing both objective (through sensors) and subjective (through a smartphone app) data collection. I have written the section on privacy issues (Section 4.5), and contributed to the abstract, introduction, challenges and conclusion sections. The smartphone app was developed by Frécon, and the backend system was developed by Johanson and Jalminger.

1.5.4 Paper IV

Generalizing the idea with sending polls to smartphones from Paper III, I wanted data collectors to be able to send any poll, without worrying about how to guarantee differential privacy. Hence, I built RANDORI, a set of tools for designing data collections and collecting data under local differential privacy. During the design phase, I let users analytically investigate the accuracy/privacy trade-off in their poll. Accuracy is estimated analytically through Chernoff

bounds. Then, I also investigate what other problems can arise during data collection, that are not captured by differential privacy itself. In particular, I identify and address side-channels that arise during data collection. My contributions are:

+ tools for designing polls and collecting data under differential privacy
+ a tool for predicting and tuning accuracy of a given poll
+ an end-to-end private implementation of randomized response in a server-client setting

1.5.5 Paper V

Intrigued by the complex relationship between RANDORI's polls (Paper IV) and error, I set out to understand error better. To be able to model arbitrarily complex relationships between parameters and error I adopt the statistical concept of *factor experiments* [45, 46, 47]. Consequently, in this paper, I introduce a novel application of factor experiments that allows us to create prediction models for the error of a differentially private algorithm. In Figure 1.7 I have visualized such a prediction model for an algorithm with two factors (parameters).

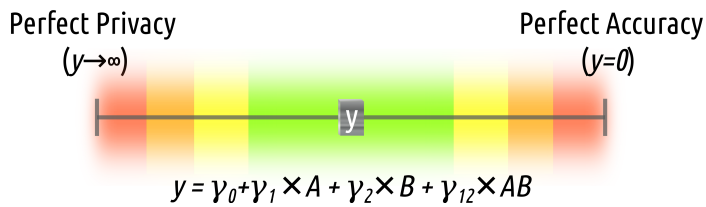


Figure 1.7: A visualization of a linear model with two factors A and B

As a use case for factor experiments, I investigate RANDORI's polls. Since RANDORI's polls can be tweaked by several parameters it is particularly interesting to understand the relationship between each parameter and error. In order to run the experiments, I extend RANDORI with a simulation environment. In

the end, I create a prediction model for error and evaluate the model's validity in the context of RANDORI's poll structure.

My contribution is:

+ A method for constructing accuracy/error prediction models

1.5.6 Paper VI

In Paper VI, we systematize the available body of research on accuracy improving techniques in differential privacy. In particular, we investigate accuracy improvement for two common data analyses: histograms and synthetic data. To accomplish our goal, we conduct a systematic literature review, categorize our findings and qualitatively summarize the results. Our main contributions are:

+ A technical summary of each algorithms in order to provide a consolidate view of the state-of-the-art.
+ Categorization that synthesize the evolutionary relationships of the research domain in differential privacy for histogram and synthetic data publication.
+ Categorization of the state-of-the-art, which is based on the conceptual relationships of the identified algorithms.

Statement of Contribution

My co-author and I have made an equal contribution to this paper. Both have contributed to methodology, SLR design, scoping, reading and analysis of papers.

1.6 Conclusion and Future Work

I started my PhD journey by conducting a systematic literature review (SLR) to understand the current landscape of security and privacy research in the context of big data analysis. This SLR resulted in Paper I. Perhaps disappointing, but

not surprising, the conclusion from Paper I was that security and privacy in big data analysis is not fundamentally different from security and privacy research in general. Still, we found that many of the papers included in the SLR worked with privacy. Here, differential privacy was overrepresented among the privacy-preserving techniques used. In fact, differential privacy actually benefits from the use of big data. To clarify, since the definition of differential privacy only puts constraints on the algorithms, and not on the data, accuracy tends to increase with the size of the data set used. As such, differential privacy seemed like a particularly interesting topic to continue to investigate.

In Paper II I investigated how differential privacy could be applied to a particular flavor of big data, namely vehicular data. Vehicular data consists of high dimensionality, time series data. At this point I had an ongoing collaboration with a vehicle manufacturer, and in Paper II I tried to address their questions and concerns about differential privacy. In hindsight, it is apparent to me that Paper II could have benefited from me suggesting the use of several methodologies I learned later on my PhD journey. For example, *principal component analysis* (PCA) and *factor experiments* are two methodologies I believe would be useful to reduce the dimensionality of vehicular data. Specifically, there exists differentially private version of PCA [48] which provides a way to convert data into fewer dimensions. As for factor experiments, they can be used to construct a linear model between variables. In other words, factor experiments help us understand the relationship between different variables in our system. As such, the linear model can be used to understand which variables are important to include in an analysis, and which ones are negligible. Consequently, I believe we could use the linear model to achieve pre-processing similar to PCA.

Additionally, it would be interesting to investigate which one of PCA and factor experiments work best in the vehicular domain. That is, is it better to 1) process the data after applying differentially private PCA, or 2) reduce the inputs based on the linear model before processing the data under differential privacy?

In parallel with my work with trying to introduce differential privacy in an automotive context, my collaborators and I investigated a particular use case. This use case included a smartphone app which would allow us to interact directly with the drivers. As such, we would be able to access a type of data that was not high dimensional, and consequently would be easier to reason about than the vehicular data itself. Still, at this point in time, we never implemented any differential private mechanism in the smartphone app. Accordingly, it would be interesting to combine the smartphone app from Paper III with RANDORI (Paper IV).

The observant reader may already have drawn this conclusion, but the idea behind RANDORI was basically to generalize the smartphone setting to a server-client setting. Moreover, we used the same data format (JSON) in the smartphone app as RANDORI. As such, it should be possible to integrate the client logic into the smartphone app with relatively little effort. Moreover, combining Paper III and Paper IV into one smartphone app opens up possibilities to conduct user studies with drivers as participants. That is, we could evaluate both the smartphone app and the effectiveness of RANDORI in one joint study. Similarly, it would be interesting to test the effectiveness of RANDORI from the data analysts' perspective.

Moving on, from Paper V, I would like to investigate if, and when, my methodology is applicable to predict accuracy in other differentially private algorithms. For example, I think it would be interesting to take open source differentially private libraries (e.g. [49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59]) and test the applicability of my methodology. Also, when working on Paper VI we came across many algorithms that have several input parameters. As such all of these algorithms would be particularly interesting to test, as having many input parameters can make it more difficult to reason about accuracy using Chernoff bounds. Specifically, I would like to investigate whether the methodology works better for algorithms with more input parameters than those with only a few. Seeing as the term e^{ε} is exponential and not linear, in which cases will my linear model provide a good fit? Moreover, in cases where a linear model does not

work, it would be interesting to extend the methodology to include non-linear models.

Lastly, in Paper VI we identified many open challenges. In particular, I think it would be interesting to combine the different accuracy improving techniques we have categorized, and see which ones actually are composable. In particular, one question that we posed but left unanswered is "*How many techniques from the same place (Figure 1.8) can we use?*". For example, is it possible combine several pre-processing techniques and gain more accuracy than when only using one pre-processing technique?

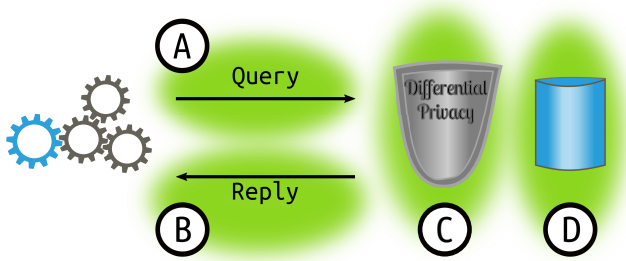


Figure 1.8: The different places we identified in Paper VI where accuracy improving techniques can be applied: (A) altering the query, (B) post-processing, (C) changes in the release mechanism, and (D) pre-processing

In addition, potential future work can be identified by looking at the family tree (Figure 1.9) we created to illustrate the relationships between the algorithms. For example, IHP and SORTaki are both descendants from AHP, indicating that they should be applicable to similar problems. As such, it might be possible to combine the ideas of both IHP and SORTaki to achieve even higher accuracy.

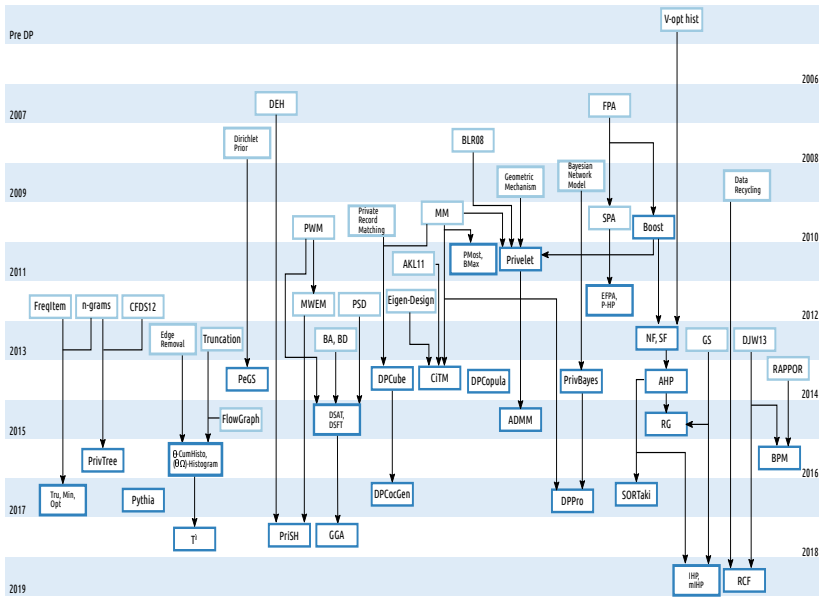


Figure 1.9: Algorithms encountered in Paper VI and their relationships

To conclude, knowing that with great data comes great responsibility, we need to make techniques that ensure data privacy accessible. Differential privacy is only one flavor of privacy-preserving techniques, and as such may not be suitable to apply in all cases. Still, differential privacy is enticing for many reasons, for example due to its rigorous mathematical foundation and its ability to protect against strong adversaries. Consequently, I have dedicated this thesis to exploring when and how differential privacy can be used in real settings. Moreover, I realized that the accuracy loss imposed by differential privacy is particularly important for practitioners to understand before they can adopt differential privacy.

Finally, I would like to add that there still exists many open challenges to address to further contribute bridge the gap between differential privacy in theory

and practice. My main interest has been to provide a better understanding of the accuracy/privacy trade-off, i.e. the balancing act, of differential privacy. In the end, the papers included in this thesis have allowed me to investigate different ways of bringing differential privacy closer to end-users by:

+ providing advice for practitioners (Paper II)
+ conducting a case study where we would like to introduce differential privacy (Paper III)
+ providing a set of tools for aiding practitioners when gathering poll data under local differential privacy (Paper IV)
+ providing a methodology to empirically evaluate the accuracy of differentially private algorithms (Paper V)
+ providing a systematization of knowledge (SoK) of accuracy improving techniques for differentially private histograms and synthetic data (Paper VI)

Bibliography

[1] United Nations. *Universal Declaration of Human Rights*. en. 1948.

[2] European Court of Human Rights and Council of Europe. *European Convention on Human Rights*. 2013.

[3] European Parliament, Council of the European Union. "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)". In: *Official Journal of the European Union* (May 2016).

[4] J. F. Skariachan Dhanya. "Target Cyber Breach Hits 40 Million Payment Cards at Holiday Peak". en. In: *Reuters* (Dec. 2013).

[5] E. A. Harris and N. Perlroth. "For Target, the Breach Numbers Grow". In: *The New York Times* (Jan. 2014).

[6] CBS and AP. *Health Insurer Anthem Hit by Hackers*. https://www.cbsnews.com/news/health-insurer-anthem-hit-by-hackers/. 2015.

[7] C. Williams. "Carphone Warehouse Hackers 'Used Traffic Bombardment Smokescreen'". In: *The Telegraph* (2015).

[8] Committee on Oversight and Government Reform. *The OPM Data Breach: How the Government Jeopardized Our National Security for*

More than a Generation. Tech. rep. House Oversight and Government Reform, 2016.

[9] Swiss Government Computer Emergency Response Team. *Technical Report about the RUAG Espionage Case*. https://www.govcert.ch/blog/technical-report-about-the-ruag-espionage-case/. 2016.

[10] D. Volz. "Yahoo Says Hackers Stole Data from 500 Million Accounts in 2014". en. In: *Reuters* (Sept. 2016).

[11] A. DeSimone and N. Horton. *Sony's Nightmare before Christmas: The 2014 North Korean Cyber Attack on Sony and Lessons for US Government Actions in Cyberspace*. en. Tech. rep. NSAD-R-17-045. The Johns Hopkins University Applied Physics Laboratory LLC, 2017, p. 44.

[12] D. Goyal. *Security Update – What Really Happened? And What Next?* 2017.

[13] S. A. O'Brien. "Giant Equifax Data Breach: 143 Million People Could Be Affected". In: *CNNMoney* (Sept. 2017).

[14] R. Jennings. *Brazil Govt's Huge Leak: Health Data of 243M*. https://securityboulevard.com/2020/12/brazil-govts-huge-leak-health-data-of-243m-people/. Dec. 2020.

[15] H. Garcia-Molina et al. *Database Systems the Complete Book*. Second. Pearson Education International, 2009.

[16] E. F. Codd. "A Relational Model of Data for Large Shared Data Banks". In: *Communications of the ACM* 13.6 (1970), pp. 377–378.

[17] D. E. Denning and P. J. Denning. "Data Security". In: *ACM Computing Surveys* 11.3 (Sept. 1979), pp. 227–249.

[18] T. Dalenius. "Towards a Methodology for Statistical Disclosure Control". In: *Statistisk Tidskrift* 15 (1977), pp. 429–444.

[19] N. E. Bordenabe and G. Smith. "Correlated Secrets in Quantitative Information Flow". en. In: *2016 IEEE 29th Computer Security Foundations Symposium (CSF)*. Lisbon: IEEE, June 2016, pp. 93–104.

[20] Denning and Schlörer. "Inference Controls for Statistical Databases". en. In: *Computer* 16.7 (July 1983), pp. 69–82.

[21] L. J. Hoffman and W. F. Miller. “Getting a personal dossier from a statistical data bank”. In: *Datamation* 16.5 (1970), pp. 74–75.

[22] L. Olsson. “Protection of Output and Stored Data in Statistical Data Bases”. In: *ADB-information* M75:4 (1975).

[23] D. E. Denning. “Secure Statistical Databases with Random Sample Queries”. en. In: *ACM Transactions on Database Systems* 5.3 (Sept. 1980), pp. 291–315.

[24] M. K. Reiter and A. D. Rubin. “Crowds: Anonymity for Web Transactions”. In: *ACM Transactions on Information and System Security* 1.1 (Nov. 1998), pp. 66–92.

[25] O. Berthold et al. “The Disadvantages of Free MIX Routes and How to Overcome Them”. en. In: *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability Berkeley, CA, USA, July 25–26, 2000 Proceedings*. Ed. by H. Federrath. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2001, pp. 30–45.

[26] M. K. Wright et al. “An Analysis of the Degradation of Anonymous Protocols”. en. In: *NDSS* 2 (2002), pp. 39–50.

[27] A. Serjantov and G. Danezis. “Towards an Information Theoretic Metric for Anonymity”. en. In: *Privacy Enhancing Technologies*. Ed. by R. Dingledine and P. Syverson. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2003, pp. 41–53.

[28] C. Diaz et al. “Towards Measuring Anonymity”. In: *Proceedings of PET* (Apr. 2002), pp. 54–68.

[29] C. Clifton and T. Tassa. “On syntactic anonymity and differential privacy”. In: *2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW)*. IEEE, 2013, pp. 88–93. URL: `http://www.computer.org/csdl/proceedings/icdew/2013/5303/00/06547433-abs.html` (visited on 07/27/2015).

[30] P. Samarati. “Protecting Respondents Identities in Microdata Release”. In: *IEEE transactions on Knowledge and Data Engineering* 13.6 (2001), pp. 1010–1027.

[31] A. Machanavajjhala et al. “L-Diversity: Privacy beyond k -Anonymity”. English. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1.1 (2007).

[32] N. Li et al. “T-Closeness: Privacy Beyond k-Anonymity and l-Diversity”. In: *ICDE ’14*. 2007.

[33] J. Cao and P. Karras. “Publishing Microdata with a Robust Privacy Guarantee”. en. In: *Proceedings of the VLDB Endowment* 5.11 (July 2012), pp. 1388–1399.

[34] C. Dwork et al. “Calibrating Noise to Sensitivity in Private Data Analysis”. In: *Theory of Cryptography*. Ed. by S. Halevi and T. Rabin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 265–284.

[35] C. Dwork. “Differential Privacy”. en. In: *Encyclopedia of Cryptography and Security*. Ed. by H. C. A. van Tilborg and S. Jajodia. Boston, MA: Springer US, 2011, pp. 338–340.

[36] C. Dwork and A. Roth. “The Algorithmic Foundations of Differential Privacy”. en. In: *Foundations and Trends® in Theoretical Computer Science* 9.3-4 (2014), pp. 211–407.

[37] S. L. Warner. “Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias”. In: *Journal of the American Statistical Association* 60.309 (Mar. 1965), pp. 63–69.

[38] J. Lee and C. Clifton. “How Much Is Enough? Choosing ε for Differential Privacy”. en. In: *Information Security*. Ed. by X. Lai et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2011, pp. 325–340.

[39] J. Hsu et al. “Differential Privacy: An Economic Method for Choosing Epsilon”. In: *2014 IEEE 27th Computer Security Foundations Symposium*. July 2014, pp. 398–410.

[40] P. Laud and A. Pankova. “Interpreting Epsilon of Differential Privacy in Terms of Advantage in Guessing or Approximating Sensitive Attributes”. In: *arXiv:1911.12777 [cs]* (Nov. 2019). arXiv: 1911.12777 [cs].

[41] D. Bernau et al. “Quantifying Identifiability to Choose and Audit ε in Differentially Private Deep Learning”. In: (Mar. 2021). arXiv: 2103.02913 [cs].

[42] A. Wood et al. “Differential Privacy: A Primer for a Non-Technical Audience”. In: *Vanderbilt Journal of Entertainment & Technology Law* 21 (2018), p. 209.

[43] C. Dwork. “Differential Privacy: A Survey of Results”. In: *Theory and Applications of Models of Computation*. Ed. by M. Agrawal et al. Lecture Notes in Computer Science 4978. Springer Berlin Heidelberg, Jan. 1, 2008, pp. 1–19.

[44] T. Zhu et al. *Differential Privacy and Applications*. Vol. 69. Advances in Information Security. Cham: Springer International Publishing, 2017.

[45] NIST/SEMATECH. *NIST/SEMATECH e-Handbook of Statistical Methods*. https://www.itl.nist.gov/div898/handbook/index.htm. [Accessed: 2021-02-17]. 2013.

[46] NIST/SEMATECH. *5.1.1. What Is Experimental Design?* https://www.itl.nist.gov/div898/handbook//pri/section1/pri11.htm. [Accessed: 2021-02-17]. 2013.

[47] NIST/SEMATECH. *5.3.3.3. Full Factorial Designs*. https://www.itl.nist.gov/div898/handbook/pri/section3/pri333.htm. [Accessed: 2021-02-24]. 2013.

[48] K. Chaudhuri et al. “A Near-Optimal Algorithm for Differentially-Private Principal Components”. en. In: *Journal of Machine Learning Research* 14 (2013), pp. 2905–2943.

[49] Differential Privacy Team. *Google/Differential-Privacy*. Google. Mar. 2021.

[50] Facebook Inc. *Opacus · Train PyTorch Models with Differential Privacy*. https://opacus.ai/. 2021.

[51] Google LLC. *Tensorflow/Privacy*. Github. Mar. 2021.

[52] N. Holohan. *Diffprivlib: IBM Differential Privacy Library*. https://github.com/IBM/differential-privacy-library. [Accessed: 2021-03-23.

[53] IBM Corp. *IBM/Discrete-Gaussian-Differential-Privacy*. https://github.com/IBM/discrete-gaussian-differential-privacy. Mar. 2021.

[54] Microsoft. *SmartNoise*. en. https://smartnoise.org. 2020.

[55] B. I. P. Rubinstein and F. Aldà. "Pain-Free Random Differential Privacy with Sensitivity Sampling". en. In: *International Conference on Machine Learning*. PMLR, July 2017, pp. 2950–2959.

[56] OpenDP. *Opendifferentialprivacy/Opendifferentialprivacy.Github.Io*. https://github.com/opendifferentialprivacy/opendifferentialprivacy.github.io. Mar. 2021.

[57] OpenMinded. *OpenMined/OM-Welcome-Package*. https://github.com/OpenMined/OM-Welcome-Package. Mar. 2021.

[58] B. Rubinstein. *Brubinstein/Diffpriv*. https://github.com/brubinstein/diffpriv. Mar. 2021.

[59] D. Zhang et al. "EKTELO: A Framework for Defining Differentially-Private Computations". In: *SIGMOD '18*. 2018.

Paper I

Boel Nelson, Tomas Olovsson

Security and Privacy for Big Data: A Systematic Literature Review

2016 IEEE International Conference on Big Data (Big Data), Washington, DC, 2016, pp. 3693-3702, doi: 10.1109/BigData.2016.7841037.

Presented at *3rd International Workshop on Privacy and Security of Big Data (PSBD 2016)*

Washington DC, USA

December 7, 2016, pp. 3693-3702.

Security and Privacy for Big Data: A Systematic Literature Review

Abstract

Big data is currently a hot research topic, with four million hits on Google scholar in October 2016. One reason for the popularity of big data research is the knowledge that can be extracted from analyzing these large data sets. However, data can contain sensitive information, and data must therefore be sufficiently protected as it is stored and processed. Furthermore, it might also be required to provide meaningful, proven, privacy guarantees if the data can be linked to individuals.

To the best of our knowledge, there exists no systematic overview of the overlap between big data and the area of security and privacy. Consequently, this review aims to explore security and privacy research within big data, by outlining and providing structure to what research currently exists. Moreover, we investigate which papers connect security and privacy with big data, and which categories these papers cover. Ultimately, is security and privacy research for big data different from the rest of the research within the security and privacy domain?

To answer these questions, we perform a *systematic literature review* (SLR), where we collect recent papers from top conferences, and categorize them in order to provide an overview of the security and privacy topics present within the context of big data. Within each category we also

present a qualitative analysis of papers representative for that specific area. Furthermore, we explore and visualize the relationship between the categories. Thus, the objective of this review is to provide a snapshot of the current state of security and privacy research for big data, and to discover where further research is required.

2.1 Introduction

Big data processing presents new opportunities due to its analytic powers. Business areas that can benefit from analyzing big data include the automotive industry, the energy distribution industry, health care and retail. Examples from these areas include analyzing driving patterns to discover anomalies in driving behaviour [1], making use of smart grid data to create energy load forecasts [2], analyzing search engine queries to detect influenza epidemics [3] and utilizing customers' purchase history to generate recommendations [4]. However, all of these examples include data linked to individuals, which makes the underlying data potentially sensitive.

Furthermore, while big data provides analytic support, big data in itself is difficult to store, manage and process efficiently due to the inherent characteristics of big data [5]. These characteristics were originally divided into three dimensions referred to as the three Vs [6], but are today often divided into four or even five Vs [2, 5, 7]. The original three Vs are *volume*, *variety* and *velocity*, and the newer V's are *veracity* and *value*. *Volume* refers to the amount of data, which Kaisler et al. [5] define to be in the range of 10^{18} bytes to be considered big data. *Variety* denotes the problem of big data being able to consist of different formats of data, such as text, numbers, videos and images. *Velocity* represents the speed at which the data grows, that is, at what speed new data is generated. Furthermore, *veracity* concerns the accuracy and trustworthiness of data. Lastly, *value* corresponds to the usefulness of data, indicating that some data points, or a combination of points, may be more valuable than others. Due to the potential large scale data processing of big data, there exists a need for efficient, scalable

solutions, that also take security and privacy into consideration.

To the best of our knowledge, there exists no peer-reviewed articles that systematically review big data papers with a security and privacy perspective. Hence, we aim to fill that gap by conducting a *systematic literature review* (SLR) of recent big data papers with a security and privacy focus. While this review does not cover the entire, vast, landscape of security and privacy for big data, it provides an insight into the field, by presenting a snapshot of what problems and solutions exists within the area.

In this paper, we select papers from top security and privacy conferences, as well as top conferences on data format and machine learning for further analysis. The papers are recent publications, published between 2012 and 2015, which we manually categorize to provide an overview of security and privacy papers in a big data context. The categories are chosen to be relevant for big data, security or privacy respectively. Furthermore, we investigate and visualize what categories relate to each other in each reviewed paper, to show what connections exists and which ones are still unexplored. We also visualize the proportion of papers belonging to each category, and the proportion of papers published in each conference. Lastly we analyze and present a representative subset of papers from each of the categories.

The paper is organized as follows. First, the method for gathering and reviewing papers is explained in Section 2.2. Then, the quantitative and qualitative results are presented in Section 2.3, where each of the categories and their corresponding papers are further analyzed in the subsection with their corresponding name. A discussion of the findings and directions for future work is presented in Section 2.4. Lastly, a conclusion follows in Section 2.5.

2.2 Methodology

In this paper, we perform a *systematic literature review* (SLR) to document what security and privacy research exists within the big data area, and identify possible areas where further research is needed. The purpose of this review is to categorize and analyze, both in a quantitative and a qualitative way, big data papers related to security or privacy. Therefore, in accordance with SLR, we define the following research questions the review should answer:

- ▷ What recent security or privacy papers exists in the big data context?
- ▷ How many papers cover security or privacy for big data?
- ▷ Which security, privacy and big data topics are represented in the area?
- ▷ When a paper covers more than one category, which categories intertwine?

SLRs originate from medical research, but has been adapted for computer science, and in particular software engineering, by Kitchenham [8] in 2004. More specifically, a SLR is useful for summarizing empirical evidence concerning an existing technology as well as for identifying gaps in current research [8]. We answer our research questions by performing the steps in the review protocol we have constructed, in accordance with Kitchenham's guidelines, displayed in Table 2.1.

1. **Data sources and search strategy:** Collect papers
2. **Study selection/study quality assessment:** Filter papers
3. **Data extraction:** Categorize papers, extract the novelty of the papers' scientific contribution
4. **Data synthesis:** Visualize papers and highlight the contributions

Table 2.1: Review protocol

As the data source, we have used papers from top conferences, ranked A^* by the Computing Research and Education Association of Australasia (CORE)[ii]

[ii] http://portal.core.edu.au/conf-ranks/

in 2014. In total, twelve relevant conferences have been chosen, including all three of CORE's top ranked security and privacy conferences. There also exists several new, promising conferences in big data. However, none of these big data specific conferences are ranked yet, and thus they are not included in this review. Arguably, the highest quality papers should appear in the A^* ranked conferences, instead of in a not proven venue. Furthermore, it is our belief that new ideas hit conferences before journals, and thus journals have been excluded from the review. Consequently, we have chosen top conferences for closely related topics: machine learning and data format[iii]. Thus, the big data conferences are represented by seven conferences from the field of data format and two from machine learning. The chosen conferences are presented in Table 2.2, and we further discuss the consequences of choosing these conferences in Section 2.4.

► **Step 1** To perform the first step from Table 2.1, the collection of papers, we have constructed the following two queries:

- **Query A:** allintitle: privacy OR private OR security OR secure
  Sources: DCC, ICDE, ICDM, SIGKDD, SIDMOD, VLDB, WSDM, ICML and NIPS
  Timespan: 2012-2015
- **Query B:** allintitle: "big data"
  Sources: DCC, ICDE, ICDM, SIGKDD, SIDMOD, VLDB, WSDM, ICML, NIPS, CCS, S&P and USENIX Security
  Timespan: 2012-2015

Acronym	Conference Name	Field(s) of Research[iv]
DCC	Data Compression Conference	Data Format
ICDE	International Conference on Data Engineering	Data Format

[iii] Field of research code 0804: http://www.abs.gov.au/Ausstats/abs@.nsf/0/206700786B8EA3EDCA257418000473E3?opendocument

[iv] As labeled by CORE

Acronym	Conference Name	Field(s) of Research[iv]
ICDM	IEEE International Conference on Data Mining	Data Format
SIGKDD	Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining	Data Format
SIGMOD	Association for Computing Machinery's Special Interest Group on Management of Data	Data Format
VLDB	International Conference on Very Large Databases	Data Format
WSDM	ACM International Conference on Web Search and Data Mining	Data Format, Distributed Computing, Library and Information Studies
ICML	International Conference on Machine Learning	Artificial Intelligence and Image Processing
NIPS	Neural Information Processing System Conference	Artificial Intelligence and Image Processing
CCS	ACM Conference on Computer and Communications Security	Computer Software
S&P	IEEE Symposium on Security and Privacy	Computation Theory and Mathematics, Computer Software
USENIX Security	Usenix Security Symposium	Computer Software

Table 2.2: Conferences the papers were collected from, including acronym and field of research

Note that only the title of a paper is used to match on a keyword. The reason for this is to reduce the amount of false positives. For example, if the search is not limited to the title, a paper might discuss the keyword in the introduction or as related work, but it might not otherwise be included in the paper. Since the review is performed manually, it would require a labor intensive analysis just to eliminate those irrelevant papers. Furthermore, we believe that the papers related to security or privacy would mention this in their title. Thus, we have focused on a smaller, relevant, subset.

Query A focuses on finding papers related to security or privacy in one of the big data conferences. This query is intentionally constructed to catch a wide range of security and privacy papers, including relevant papers that have omitted 'big data' from the title. Furthermore, query B is designed to find big data papers in any of the conferences, unlike query A. The reason to also include query B is foremost to capture big data papers in security and privacy conferences. Query B will also be able to find big data papers in the other conferences, which provides the opportunity to catch security or privacy papers that were not already captured by query A.

▶ **Step 2** After the papers have been collected, we manually filter them to perform both a selection and a quality assessment, in accordance with the guidelines for a SLR. First, we filter away talks, tutorials, panel discussions and papers only containing abstracts from the collected papers. We also verify that no papers are duplicates to ensure that the data is not skewed. Then, as a quality assessment we analyze the papers' full corpora to determine if they belong to security or privacy. Papers that do not discuss security or privacy are excluded. Thus, the irrelevant papers, mainly captured by query B, and other potential false positives, are eliminated.

To further assess the quality of the papers, we investigate each papers' relevance for big data. To determine if it is a big data paper we include the entire corpus of the paper, and look for evidence of scalability in the proposed solution by examining if the paper relates to the five V's. The full list of included and ex-

cluded papers is omitted in this paper due to space restrictions, but it is available from the authors upon request.

► **Step 3** Then, each paper is categorized into one or more of the categories shown in Table 2.3. These categories were chosen based on the five V's, with additional security and privacy categories added to the set. Thus the categories capture both the inherent characteristics of big data, as well as security and privacy.

Category	V	Security or Privacy
Confidentiality[iv]		✓
Data Analysis	Value	
Data Format	Variety, Volume	
Data Integrity	Veracity	✓
Privacy[v]		✓
Stream Processing	Velocity, Volume	
Visualization	Value, Volume	

Table 2.3: Categories used in the review, chosen based on the five V's. A checkmark in the third column means that the category is a security or privacy category.

In total, 208 papers match the search criteria when we run both queries in Google Scholar. After filtering away papers and performing the quality assessment, 82 papers remain. Query A results in 78 papers, and query B contributes with four unique papers that were not already found by query A. In Table 2.4 the number of papers from each conference is shown for query A and query B respectively.

[iv] As defined by ISO 27000:2016 [9]
[v] *Anonymization* as defined by ISO 29100:2011 [10]

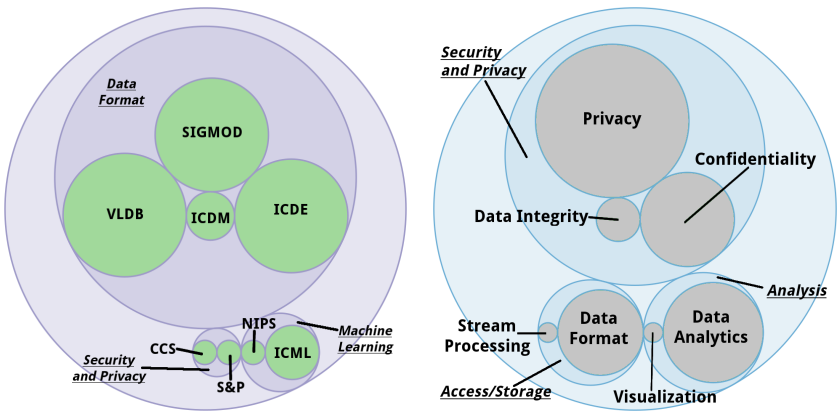
Conference Acronym	Query A		Query B	
	Number of Papers	Percentage of Papers	Number of Papers	Percentage of Papers
DCC	0	0%	0	0%
ICDE	22	28%	0	0%
ICDM	4	5%	0	0%
SIGKDD	0	0%	0	0%
SIGMOD	21	26%	1	25%
VLDB	25	31%	1	25%
WSDM	0	0%	0	0%
ICML	5	6.3%	0	0%
NIPS	1	1.3%	0	0%
S&P	-	-	1	25%
USENIX Security	-	-	0	0%
CCS	-	-	1	25%
Total:	78	100%	4	100%

Table 2.4: The number, and percentage, of papers picked from each conference, for query A and query B

► **Step 4** Then, as part of the data synthesis which is the last step in the review protocol in Table 2.1, the quantitative results from the queries are visualized. Both as circle packing diagrams, where the proportion of papers and conferences is visualized, and as a circular network diagram where relationships between categories are visualized. Thereafter a qualitative analysis is performed on the papers, where the novel idea and the specific topics covered are extracted from the papers' corpora. A representative set of the papers are then presented.

2.3 Results

In this section, we quantitatively and qualitatively analyze the 82 papers. Figure 2.1 (a) visualizes where each paper originates from, using circle packing diagrams. The size of each circle corresponds to the proportion of papers picked from a conference. As can be seen, most papers have been published in ICDE, SIGMOD or VLDB. Furthermore, the distribution of the different categories is illustrated in Figure 2.1 (b), where the size of a circle represents the amount of papers covering that category. Prominent categories are *privacy*, *data analysis* and *confidentiality*.



(a) Conferences, grouped by research field (b) Categories, grouped by similarity

Figure 2.1: Circle packing diagrams, showing the proportion of papers belonging to conferences (a) and categories (b)

Furthermore, some papers discuss more than one category and therefore belong to more than one category. Therefore, the total number of papers when all categories are summed will exceed 82. To illustrate this overlap of categories, the relationship between the categories is visualized as a circular network diagram in Figure 2.2. Each line between two categories means that there exists at least one paper that discusses both categories. The thickness of the line reflects the

amount of papers that contain the two categories connected by the line. *Privacy* and *data analytics* as well as *confidentiality* and *data format* are popular combinations. *Stream processing* and *visualization* are only connected by one paper, respectively, to *privacy*.

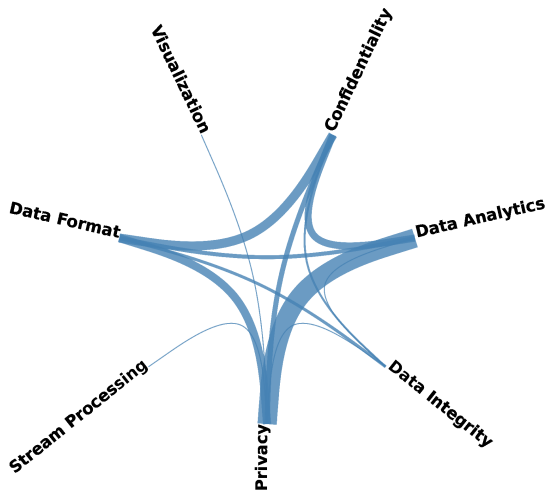


Figure 2.2: Connections between categories, where the thickness of the link represents the amount of papers that connect the two categories

Since there is not enough room to describe each paper in the qualitative analysis, we have chosen a representative set for each category. This representative set is chosen to give an overview of the papers for each category. Each selected paper is then presented in a table to show which categories it belongs to. An overview of the rest of the papers are shown in Table 2.5.

Author	Short Title	C	A	DF	DI	P	SP	V
Akcora et al.	Privacy in Social Networks					✓		
Allard et al.	Chiaroscuro	✓	✓			✓		

Author	Short Title	C	A	DF	DI	P	SP	V
Bonomi and Xiong	Mining Frequent Patterns with Differential Privacy		✓			✓		
Bonomi et al.	LinkIT					✓		
Cao et al.	A hybrid private record linkage scheme	✓				✓		
Chen and Zhou	Recursive Mechanism			✓		✓		
Dev	Privacy Preserving Social Graphs for High Precision Community Detection		✓			✓		
Dong et al.	When Private Set Intersection Meets Big Data	✓						
Fan et al.	FAST					✓		
Gaboardi et al.	Dual Query					✓		
Guarnieri and Basin	Optimal Security-aware Query Processing	✓		✓				
Guerraoui et al.	D2P					✓		
Haney et al.	Design of Policy-aware Differentially Private Algorithms					✓		
He et al.	Blowfish Privacy					✓		
He et al.	DPT		✓			✓		
He et al.	SDB	✓		✓				

Author	Short Title	C	A	DF	DI	P	SP	V
Hu et al.	Authenticating Location-based Services Without Compromising Location Privacy	✓				✓		
Hu et al.	Private search on key-value stores with hierarchical indexes	✓						
Hu et al.	VERDICT	✓		✓				
Jain and Thakurta	(Near) Dimension Independent Risk Bounds for Differentially Private Learning		✓			✓		
Jorgensen and Cormode	Conservative or liberal?					✓		
Kellaris and Papadopoulos	Practical differential privacy via grouping and smoothing					✓		
Khayyat et al.	BigDansing			✓	✓			
Kozak and Zezula	Efficiency and Security in Similarity Cloud Services	✓				✓		
Li and Miklau	An Adaptive Mechanism for Accurate Query Answering Under Differential Privacy					✓		

Author	Short Title	C	A	DF	DI	P	SP	V
Li et al.	A Data- and Workload-aware Algorithm for Range Queries Under Differential Privacy					✓		
Li et al.	DPSynthesizer					✓		
Li et al.	Fast Range Query Processing with Strong Privacy Protection for Cloud Computing					✓		
Li et al.	PrivBasis		✓			✓		
Lin and Kifer	Information Preservation in Statistical Privacy and Bayesian Estimation of Unattributed Histograms					✓		
Lu et al.	Generating private synthetic databases for untrusted system evaluation			✓		✓		
Mohan et al.	GUPT		✓			✓		
Nock et al.	Rademacher observations, private data, and boosting		✓			✓		
Oktay et al.	SEMROD	✓		✓				
Pattuk et al.	Privacy-aware dynamic feature selection		✓			✓		

Author	Short Title	C	A	DF	DI	P	SP	V
Potluru et al.	CometCloudCare (C3)		✓	✓		✓		
Qardaji et al.	Differentially private grids for geospatial data					✓		
Qardaji et al.	PriView					✓		
Qardaji et al.	Understanding Hierarchical Methods for Differentially Private Histograms					✓		
Rahman et al.	Privacy Implications of Database Ranking					✓		
Rana et al.	Differentially Private Random Forest with High Utility		✓			✓		
Ryu et al.	Curso					✓		
Sen et al.	Bootstrapping Privacy Compliance in Big Data Systems	✓						
Shen and Jin	Privacy-Preserving Personalized Recommendation					✓		
Terrovitis et al.	Privacy Preservation by Disassociation					✓		
To et al.	A Framework for Protecting Worker Location Privacy in Spatial Crowdsourcing					✓		

Author	Short Title	C	A	DF	DI	P	SP	V
Wong et al.	Secure Query Processing with Data Interoperability in a Cloud Database Environment	✓		✓				
Xiao et al.	DPCube			✓		✓		
Xu et al.	Differentially private frequent sequence mining via sampling-based candidate pruning		✓			✓		
Xue et al.	Destination prediction by sub-trajectory synthesis and privacy protection against such prediction		✓			✓		
Yang et al.	Bayesian Differential Privacy on Correlated Data					✓		
Yaroslavtsev et al.	Accurate and efficient private release of datacubes and contingency tables					✓		
Yi et al.	Practical k nearest neighbor queries with location privacy	✓	✓					
Yuan et al.	Low-rank Mechanism					✓		

Author	Short Title	C	A	DF	DI	P	SP	V
Zeng et al.	On Differentially Private Frequent Itemset Mining		✓			✓		
Zhang et al.	Functional Mechanism		✓			✓		
Zhang et al.	Lightweight privacy-preserving peer-to-peer data integration	✓						
Zhang et al.	Private Release of Graph Statistics Using Ladder Functions					✓		
Zhang et al.	PrivBayes					✓		
Zhang et al.	PrivGene		✓			✓		

Table 2.5: The reviewed papers omitted from the reference list, showing categories covered by each paper. C = Confidentiality, DA = Data Analysis, DF = Data Format, DI= Data Integrity, P = Privacy, SP = Stream Processing, V = Visualization.

2.3.1 Confidentiality

Confidentiality is a key attribute to guarantee when sensitive data is handled, especially since being able to store and process data while guaranteeing confidentiality could be an incentive to get permission to gather data. In total, 23 papers were categorized as confidentiality papers. Most papers used different types of encryption, but there was no specific topic that had a majority of papers. Instead, the papers were spread across a few different topics. In Table 2.6, an overview of all papers presented in this section is given.

Five papers use *homomorphic encryption*, which is a technique that allows certain arithmetic operations to be performed on encrypted data. Of those five papers, one uses fully homomorphic encryption which supports any arithmetic operation, whereas the rest use partial homomorphic encryption which supports given arithmetic operations. Liu et al. [11] propose a secure method for comparing trajectories, for example to compare different routes using GPS data, by using partial homomorphic encryption. Furthermore, Chu et al. [12] use fully homomorphic encryption to provide a protocol for similarity ranking.

Another topic covered by several papers is *access control*. In total, four papers discuss access control. For example, Bender et al. [13] proposed a security model where policies must be explainable. By explainable in this setting Bender et al. refers to the fact that every time a query is denied due to missing privileges, an explanation as to what additional privileges are needed is returned. This security model is an attempt to make it easier to implement the principle of least privilege, rather than giving users too generous privileges. Additionally, Meacham and Shasha [14] propose an application that provides access control in a database, where all records are encrypted if the user does not have the appropriate privileges. Even though the solutions by Bender et al. and Meacham and Shasha use SQL, traditionally not associated with big data, their main ideas are still applicable since it only requires changing the database to a RDBMS for big data that have been proposed earlier, such as Vertica [15] or Zhu et al.'s [16] distributed query engine.

Other topics covered were *secure multiparty computation*, a concept where multiple entities perform a computation while keeping each entity's input confidential, *oblivious transfer*, where a sender may or may not transfer a piece of information to the receiver without knowing which piece is sent, as well as different *encrypted indexes* used for improving search time efficiency. In total, three papers use secure multiparty computation, two use oblivious transfer and two use encrypted indexes.

Author	C	DA	DF	DI	P	SP	V
Bender et al. [13]	✓						
Chu et al. [12]	✓	✓	✓				
Liu et al. [11]	✓	✓					
Meacham and Shasha [14]	✓	✓					

Table 2.6: A set of confidentiality papers, showing categories covered by each paper. A checkmark indicates the paper on that row contains the category.

2.3.2 Data Integrity

Data integrity is the validity and quality of data. It is therefore strongly connected to veracity, one of the five V's. In total, five papers covered data integrity. Since there is only a small set of data integrity papers, no apparent topic trend was spotted. Nonetheless, one paper shows an *attack* on integrity, two papers are on *error correction and data cleansing* and two papers use *tamper-proof hardware* to guarantee integrity of the data. An overview of all papers covered in this section are shown in Table 2.7.

Xiao et al. [17] shows that it is enough to poison 5% of the training values, a data set used solely to train a machine learning algorithm, in order for feature selection to fail. Feature selection is the step where relevant attributes are being decided, and it is therefore an important step since the rest of the algorithm will depend on these features. Thus, Xiao et al. show that feature selection is not secure unless the integrity of the data can be verified.

Furthermore, Arasu et al. [18] implemented a SQL database called Cipherbase that focuses on confidentiality of data as well as integrity in the cloud. To maintain the integrity of the cryptographic keys, they use FPGA based custom hardware to provide tamper-proof storage. Lallali et al. [19] also used tamper-resistant hardware where they enforce confidentiality for queries performed in personal clouds. The tamper-resistant hardware is in the form of a secure to-

ken which prevents any data disclosure during the execution of a query. While the secure tokens ensures a closed execution environment, they posses limited processing power due to the hardware constraints which adds to the technical challenge.

Author	C	DA	DF	DI	P	SP	V
Arasu et al. [18]	✓		✓	✓			
Lallali et al. [19]	✓			✓	✓		
Xiao et al. [17]		✓		✓			

Table 2.7: A set of data integrity papers, showing categories covered by each paper

2.3.3 Privacy

An important notion is privacy for big data, since it can potentially contain sensitive data about individuals. To mitigate the privacy problem, data can be de-identified by removing attributes that would identify an individual. This is an approach that works, if done correctly, both when data is managed and when released. However, under certain conditions it is still possible to re-identify individuals even when some attributes have been removed [20, 21, 22]. Lu et al. [7] also point out that the risk of re-identification can increase with big data, as more external data from other sources than the set at hand can be used to cross-reference and infer additional information about individuals.

Several privacy models, such as k-anonymity [23], l-diversity [24], t-closeness [25] and differential privacy [26], can be used to anonymize data. The first three are techniques for releasing entire sets of data through privacy-preserving data publishing (PPDP), whereas differential privacy is used for privacy-preserving data mining (PPDM). Thus, differential privacy is obtained without processing the entire data set, unlike the others. Therefore, anonymizing larger data sets can be difficult from an efficiency perspective. How-

ever, larger sets have greater potential to hide individual data points within the set [27].

Out of a total of 61 privacy papers, one paper [28] uses *k-anonymity*, and another paper [29] uses *l-diversity* and *t-closeness* but also *differential privacy* to anonymize data. Furthermore, Cao and Karras [30] introduce a successor to *t*-closeness, called β*-likeness* which they claim is more informative and comprehensible. In comparison, a large portion, 46 papers, of the privacy oriented papers focuses only on differential privacy as their privacy model. Most of them propose methods for releasing differentially private data structures. Among these are differentially private histograms [31] and different data structures for differentially private multidimensional data [32].

An interesting observation by Hu et al. [33] is that differential privacy can have a large impact on accuracy of the result. When Hu et al. enforced differential privacy on their telecommunications platform, they got between 15% to 30% accuracy loss. In fact, guaranteeing differential privacy while maintaining high utility of the data is not trivial. From the reviewed papers, 15 of them investigated utility in combination with differential privacy.

One example of a paper that investigates the utility of differentially private results, and how to improve it is Proserpio et al. [34]. The work of Proserpio et al. is a continuation of the differentially private querying language PINQ [35], which they enhance by decreasing the importance of challenging entries, which induce high noise, in order to improve accuracy of the results.

The papers reviewed in this section can be seen in Table 2.8.

2.3.4 Data Analysis

Data analysis is the act of extracting knowledge from data. It includes both general algorithms for knowledge discovery, and machine learning. Out of 26

Author	C	DA	DF	DI	P	SP	V
Acs et al.[31]					✓		
Cao and Karras [30]					✓		
Cormode et al.[32]					✓		
Hu et al. [33]			✓		✓		
Jurczyk et al. [29]			✓		✓		
Proserpio et al. [34]			✓		✓		
Wang and Zheng [28]					✓		

Table 2.8: A set of privacy papers, showing categories covered by each paper

papers categorized as data analysis papers, 15 use *machine learning*. Apart from machine learning, other topics included *frequent sequence mining*, where reoccurring patterns are detected, and different versions of the *k-nearest neighbor (kNN) algorithm*, that finds the k closest points given a point of reference. All papers from this section are shown in Table 2.9.

Jain and Thakurta [36] implemented differentially private learning using kernels. The problem investigated by Jain and Thakurta is keeping the features, which are different attributes of an entity, of a learning set private while still providing useful information.

Furthermore, Elmehdwi et al. [37] implemented a secure kNN algorithm, based on partial homomorphic encryption. Here, Elmehdwi et al. propose a method for performing kNN in the cloud, where both the query and the database are encrypted. Similarly, Yao et al. [38] investigated the secure nearest neighbour (SNN) problem which asks a third party to find the point closest to a given point, without revealing any of the points to the third party. They show attacks for existing methods for SNN, and design a new SNN method that withstand the attacks.

Author	C	DA	DF	DI	P	SP	V
Elmehdwi et al. [37]	✓	✓	✓				
Jain and Thakurta [36]		✓			✓		
Yao et al. [38]	✓	✓					

Table 2.9: A set of data analysis papers, showing categories covered by each paper

2.3.5 Visualization

Visualization of big data provides a quick overview of the data points. It is an important technique, especially while exploring a new data set. However, it is not trivial to implement for big data. Gordov and Gubarev [39] point out visual noise, large image perception, information loss, high performance requirements and high rate of image change as the main challenges when visualizing big data.

One paper, by To et al. [40], shown in Table 2.10, was categorized as a visualization paper. To et al. implemented a toolbox for visualizing and assigning tasks based on an individuals' location. In this toolbox, location privacy is provided while at the same time allowing for allocation strategies of tasks to be analyzed. Thus, it presents a privacy-preserving way of analyzing how parameters in a system should be tuned to result in a satisfactory trade-off between privacy and accuracy.

Author	C	DA	DF	DI	P	SP	V
To et al. [40]		✓			✓		✓

Table 2.10: All visualization papers, showing categories covered by each paper

2.3.6 Stream Processing

Stream processing is an alternative to the traditional store-then-process approach, which can allow processing of data in real-time. The main idea is to perform analysis on data as it is being gathered, to directly address the issue of data velocity. Processing streamed data also allows an analyst to only save the results from the analysis, thus requiring less storage capacity in comparison with saving the entire data set. Furthermore, stream processing can also completely remove the bottleneck of first writing data to disk and then reading it back in order to process it if it is carried out in real-time.

One paper, by Kellaris et al. [41] shown in Table 2.11, combines stream processing with a privacy, and provides a differentially private way of querying streamed data. Their approach enforces w event-level based privacy rather than user-level privacy, which makes each event in the stream private, rather than the user that continuously produces events. Event-level based privacy, originally introduced by Dwork et al. [42], is more suitable in this case due to the fact that differential privacy requires the number of queries connected to the same individual to be known in order to provide user-level based privacy. In the case of streaming however, data is gathered continuously, making it impossible to estimate how many times a certain individual will produce events in the future.

Author	C	DA	DF	DI	P	SP	V
Kellaris et al. [41]					✓	✓	

Table 2.11: All stream processing papers, showing categories covered by each paper

2.3.7 Data Format

In order to store and access big data, it can be structured in different ways. Out of the 19 papers labeled as data format papers, most used a *distributed file system*, *database* or *cloud* that made them qualify in this category. An overview of all papers from this section can be found in Table 2.12.

One example of combining data format and privacy is the work by Peng et al. [43] that focuses on query optimization under differential privacy. The main challenge faced when enforcing differential privacy on databases is the interactive nature of the database where new queries are issued in real-time. An unspecified number of queries makes it difficult to wisely spend the privacy budget, which essentially keeps track of how many queries can be asked, used to guarantee differential privacy, to still provide high utility of query answers. Therefore, Peng et al. implemented the query optimizer Pioneer, that makes use of old query replies when possible in order to consume as little as possible of the remaining privacy budget.

Furthermore, Sathiamoorthy et al. [44] focus on data integrity, and present an alternative to standard Reed-Solomon codes, which are erasure codes used for error-correction, that are more efficient and offer higher reliability. They implemented their erasure codes in the Hadoop's distributed file system, HDFS, and were able to show that the network traffic could be reduced, but instead their erasure codes required more storage space than traditional Reed-Solomon codes.

Lastly, Wang and Ravishankar [45] point out that providing both efficient and confidential queries in databases is challenging. Inherently, the problem stems from the fact that indexes invented to increase performance of queries also leak information that can allow adversaries to reconstruct the plaintext, as Wang and Ravishankar show. Consequently, Wang and Ravishankar present an encrypted index that provides both confidentiality and efficiency for range queries, tackling the usual trade-off between security and performance.

Author	C	DA	DF	DI	P	SP	V
Peng et al. [43]			✓		✓		
Sathiamoorthy et al. [44]			✓	✓			
Wang and Ravishankar [45]	✓		✓				

Table 2.12: A set of data format papers, showing categories covered by each paper

2.4 Discussion and Future Work

While this review investigates security and privacy for big data, it does not cover all papers available within the topic, since it would be infeasible to manually review them all. Instead, the focus of this review is to explore recent papers and to provide both a qualitative and a quantitative analysis, in order to create a snapshot of the current state-of-the-art. By selecting papers from top conferences and assessing their quality manually before selecting them, we include only papers relevant for big data, security and privacy.

A potential problem with only picking papers from top conferences is that, while the quality of the papers is good, the conferences might only accept papers with ground breaking ideas. After conducting this review, however, we believe most big data solutions with respect to security and privacy are not necessarily ground breaking ideas, but rather new twists on existing ideas. From the papers collected for this review, none of the topics covered are specific for big data, rather the papers present new combinations of existing topics. Thus, it seems that security and privacy for big data is not different from other security and privacy research, as the ideas seem to scale well.

Another part of the methodology that can be discussed is the two queries used to collect papers. Query A was constructed to cover a wide range of papers, and query B was set to only include big data papers. Unfortunately, query A contributed with far more hits than query B after the filtering step from Table 2.1.

This means that most papers might not have been initially intended for big data, but they were included after the quality assessment step, since the methods used were deemed scalable. Consequently, widening the scope of query B might include papers that present security or privacy solutions solely intended for big data.

Regarding the categories, *confidentiality* was covered by almost a third of the papers, but had no dominating topic. Rather, it contained a wide spread of different cryptographic techniques and access control. Furthermore, *privacy* was well represented, with 61 papers in the review. A large portion of these papers used differential privacy, the main reason probably being the fact that most differentially private algorithms are independent of the data set's size, which makes it beneficial for large data sets.

While *privacy* was covered by a large portion of papers, only two papers use an existing privacy-preserving data publishing (PPDP) technique. Moreover, one paper introduces a new PPDP technique called β-likeness. A reason for why this topic might not be getting a lot of attention is the fact that PPDP is dependent on the size of the data set. Thus PPDP is harder to apply to big data, since the entire data set must be processed in order to anonymize it. Consequently, further work may be required in this area to see how PPDP can be applied to big data.

We have also detected a gap in the knowledge considering *stream processing* and *visualization* in combination with either *data integrity* or *confidentiality*, as no papers covered two of these topics. *Data integrity* is also one of the topics that were underrepresented, with five papers out of 82 papers in total, which is significantly lower than the number of *confidentiality* and *privacy* papers. However, it might be explained by the fact that the word 'integrity' was not part of any of the queries. This is a possible expansion of the review.

2.5 Conclusion

There are several interesting ideas for addressing security and privacy issues within the context of big data. In this paper, 208 recent papers have been collected from A^* conferences, to provide an overview of the current state-of-the-art. In the end, 82 were categorized after passing the filtering and quality assessment stage. All reviewed papers can be found in tables in Section 2.3.

Conclusively, since papers can belong to more than one category, 61 papers investigate *privacy*, 25 *data analysis*, 23 *confidentiality*, 19 *data format*, 5 *data integrity*, one *stream processing* and one *visualization*. Prominent topics were *differential privacy*, *machine learning* and *homomorphic encryption*. None of the identified topics are unique for big data.

Categories such as *privacy* and *data analysis* are covered in a large portion of the reviewed papers, and 20 of them investigate the combination of *privacy* and *data analysis*. However, there are certain categories where interesting connections could be made that do not yet exist. For example, one combination that is not yet represented is *stream processing* with either *confidentiality* or *data integrity*. *Visualization* is another category that was only covered by one paper.

In the end, we find that the security and privacy for big data, based on the reviewed papers, is not different from security and privacy research in general.

2.5 Acknowledgements

This research was sponsored by the BAuD II project (2014-03935) funded by VINNOVA, the Swedish Governmental Agency for Innovation Systems.

Bibliography

[1] G. Fuchs et al. “Constructing semantic interpretation of routine and anomalous mobility behaviors from big data”. In: *SIGSPATIAL Special* 7.1 (May 2015), pp. 27–34.

[2] M. Chen et al. “Big Data: A Survey”. en. In: *Mobile Networks and Applications* 19.2 (Jan. 2014), pp. 171–209.

[3] J. Ginsberg et al. “Detecting influenza epidemics using search engine query data”. English. In: *Nature* 457.7232 (Feb. 2009), pp. 1012–4.

[4] O. Tene and J. Polonetsky. “Privacy in the Age of Big Data: A Time for Big Decisions”. In: *Stanford Law Review Online* 64 (Feb. 2012), p. 63.

[5] S. Kaisler et al. “Big Data: Issues and Challenges Moving Forward”. English. In: *System Sciences (HICSS), 2013 46th Hawaii International Conference on*. IEEE, Jan. 2013, pp. 995–1004.

[6] D. Laney. *3D Data Management: Controlling Data Volume, Velocity, and Variety*. Tech. rep. META Group, Feb. 2001.

[7] R. Lu et al. “Toward efficient and privacy-preserving computing in big data era”. English. In: *Network, IEEE* 28.4 (Aug. 2014), pp. 46–50.

[8] B. Kitchenham. *Procedures for performing systematic reviews*. Joint Technical Report. Keele, UK: Software Engineering Group Department of Computer Science Keele University, UK, and Empirical Software Engineering, National ICT Australia Ltd, 2004, p. 26.

[9] International Organization for Standardization. *Information technology – Security techniques – Information security management systems – Overview and vocabulary*. Standard. Geneva, CH: International Organization for Standardization, Feb. 2016.

[10] International Organization for Standardization. *Information technology – Security techniques – Privacy framework*. Standard. Geneva, CH: International Organization for Standardization, Dec. 2011.

[11] A. Liu et al. "Efficient secure similarity computation on encrypted trajectory data". In: *2015 IEEE 31st International Conference on Data Engineering (ICDE)*. 2015 IEEE 31st International Conference on Data Engineering (ICDE). 2015, pp. 66–77.

[12] Y.-W. Chu et al. "Privacy-Preserving SimRank over Distributed Information Network". In: *2012 IEEE 12th International Conference on Data Mining (ICDM)*. 2012 IEEE 12th International Conference on Data Mining (ICDM). 2012, pp. 840–845.

[13] G. Bender et al. "Explainable Security for Relational Databases". In: *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*. SIGMOD '14. New York, NY, USA: ACM, 2014, pp. 1411–1422.

[14] A. Meacham and D. Shasha. "JustMyFriends: Full SQL, Full Transactional Amenities, and Access Privacy". In: *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. SIGMOD '12. New York, NY, USA: ACM, 2012, pp. 633–636.

[15] C. Bear et al. "The vertica database: SQL RDBMS for managing big data". In: *Proceedings of the 2012 workshop on Management of big data systems*. ACM, 2012, pp. 37–38.

[16] F. Zhu et al. "A Fast and High Throughput SQL Query System for Big Data". In: *Web Information Systems Engineering - WISE 2012*. Ed. by X. S. Wang et al. Lecture Notes in Computer Science 7651. DOI: 10.1007/978-3-642-35063-4_66. Springer Berlin Heidelberg, 2012, pp. 783–788.

[17] H. Xiao et al. “Is Feature Selection Secure against Training Data Poisoning?” In: *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*. 2015, pp. 1689–1698.

[18] A. Arasu et al. “Secure Database-as-a-service with Cipherbase”. In: *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. SIGMOD ’13. New York, NY, USA: ACM, 2013, pp. 1033–1036.

[19] S. Lallali et al. “A Secure Search Engine for the Personal Cloud”. In: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. SIGMOD ’15. New York, NY, USA: ACM, 2015, pp. 1445–1450.

[20] M. Barbaro and T. Zeller. “A Face Is Exposed for AOL Searcher No. 4417749”. In: *The New York Times* (Aug. 2006).

[21] A. Narayanan and V. Shmatikov. “Robust De-anonymization of Large Sparse Datasets”. In: *IEEE Symposium on Security and Privacy, 2008. SP 2008*. May 2008, pp. 111–125.

[22] P. Samarati and L. Sweeney. *Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression*. Tech. rep. SRI International, 1998.

[23] P. Samarati. “Protecting Respondents Identities in Microdata Release”. In: *IEEE transactions on Knowledge and Data Engineering* 13.6 (2001), pp. 1010–1027.

[24] A. Machanavajjhala et al. “L-diversity: Privacy beyond k -anonymity”. In: *ACM Transactions on Knowledge Discovery from Data* 1.1 (2007), 3–es.

[25] N. Li et al. “t-Closeness: Privacy Beyond k-Anonymity and l-Diversity.” In: *ICDE*. Vol. 7. 2007, pp. 106–115.

[26] C. Dwork. “Differential Privacy”. en. In: *Automata, Languages and Programming*. Ed. by M. Bugliesi et al. Lecture Notes in Computer Science 4052. Springer Berlin Heidelberg, Jan. 2006, pp. 1–12.

[27] H. Zakerzadeh et al. “Privacy-preserving big data publishing”. In: *Proceedings of the 27th International Conference on Scientific and Statistical Database Management*. ACM, June 2015, p. 26.

[28] Y. Wang and B. Zheng. “Preserving privacy in social networks against connection fingerprint attacks”. In: *2015 IEEE 31st International Conference on Data Engineering (ICDE)*. 2015 IEEE 31st International Conference on Data Engineering (ICDE). 2015, pp. 54–65.

[29] P. Jurczyk et al. “DObjects+: Enabling Privacy-Preserving Data Federation Services”. In: *2012 IEEE 28th International Conference on Data Engineering (ICDE)*. 2012 IEEE 28th International Conference on Data Engineering (ICDE). 2012, pp. 1325–1328.

[30] J. Cao and P. Karras. “Publishing Microdata with a Robust Privacy Guarantee”. In: *Proc. VLDB Endow.* 5.11 (2012), pp. 1388–1399.

[31] G. Acs et al. “Differentially Private Histogram Publishing through Lossy Compression”. In: *2012 IEEE 12th International Conference on Data Mining (ICDM)*. 2012 IEEE 12th International Conference on Data Mining (ICDM). 2012, pp. 1–10.

[32] G. Cormode et al. “Differentially Private Spatial Decompositions”. In: *2012 IEEE 28th International Conference on Data Engineering (ICDE)*. 2012 IEEE 28th International Conference on Data Engineering (ICDE). 2012, pp. 20–31.

[33] X. Hu et al. “Differential Privacy in Telco Big Data Platform”. In: *Proc. VLDB Endow.* 8.12 (2015), pp. 1692–1703.

[34] D. Proserpio et al. “Calibrating Data to Sensitivity in Private Data Analysis: A Platform for Differentially-private Analysis of Weighted Datasets”. In: *Proc. VLDB Endow.* 7.8 (2014), pp. 637–648.

[35] F. 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*. ACM, 2009, pp. 19–30.

[36] P. Jain and A. Thakurta. “Differentially private learning with kernels”. In: *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*. 2013, pp. 118–126.

[37] Y. Elmehdwi et al. “Secure k-nearest neighbor query over encrypted data in outsourced environments”. In: *2014 IEEE 30th International Conference on Data Engineering (ICDE)*. 2014 IEEE 30th International Conference on Data Engineering (ICDE). 2014, pp. 664–675.

[38] B. Yao et al. “Secure nearest neighbor revisited”. In: *2013 IEEE 29th International Conference on Data Engineering (ICDE)*. 2013 IEEE 29th International Conference on Data Engineering (ICDE). 2013, pp. 733–744.

[39] E. Y. Gorodov and V. V. Gubarev. “Analytical review of data visualization methods in application to big data”. In: *Journal of Electrical and Computer Engineering* 2013 (Jan. 2013), p. 22.

[40] H. To et al. “PrivGeoCrowd: A toolbox for studying private spatial Crowdsourcing”. In: *2015 IEEE 31st International Conference on Data Engineering (ICDE)*. 2015 IEEE 31st International Conference on Data Engineering (ICDE). 2015, pp. 1404–1407.

[41] G. Kellaris et al. “Differentially Private Event Sequences over Infinite Streams”. In: *Proc. VLDB Endow.* 7.12 (2014), pp. 1155–1166.

[42] C. Dwork et al. “Differential privacy under continual observation”. In: Proceedings of the forty-second ACM symposium on Theory of computing. ACM, 2010, pp. 715–724. (Visited on 07/12/2016).

[43] S. Peng et al. “Query optimization for differentially private data management systems”. In: *2013 IEEE 29th International Conference on Data Engineering (ICDE)*. 2013 IEEE 29th International Conference on Data Engineering (ICDE). 2013, pp. 1093–1104.

[44] M. Sathiamoorthy et al. “XORing elephants: novel erasure codes for big data”. In: *Proceedings of the 39th international conference on Very Large Data Bases*. VLDB’13. Trento, Italy: VLDB Endowment, 2013, pp. 325–336.

[45] P. Wang and C. V. Ravishankar. “Secure and efficient range queries on outsourced databases using Rp-trees”. In: *2013 IEEE 29th International Conference on Data Engineering (ICDE)*. 2013 IEEE 29th International Conference on Data Engineering (ICDE). 2013, pp. 314–325.

Paper II

Boel Nelson, Tomas Olovsson

Introducing Differential Privacy to the Automotive Domain: Opportunities and Challenges

2017 IEEE 86th Vehicular Technology Conference (VTC-Fall), Toronto, ON, 2017, pp. 1-7, doi: 10.1109/VTCFall.2017.8288389.

Presented at *2nd International Workshop on Vehicular Security (V-SEC 2017)*

Toronto, Canada

September 24, 2017

Introducing Differential Privacy to the Automotive Domain: Opportunities and Challenges

Abstract

Privacy research is attracting increasingly more attention, especially with the upcoming general data protection regulation (GDPR) which will impose stricter rules on storing and managing personally identifiable information (PII) in Europe. For vehicle manufacturers, gathering data from connected vehicles presents new analytic opportunities, but if the data also contains PII, the data comes at a higher price when it must either be properly *de-identified* or gathered with contracted consent from the drivers.

One option is to establish contracts with every driver, but the more tempting alternative is to simply de-identify data before it is gathered, to avoid handling PII altogether. However, several real-world examples have previously shown cases where *re-identification* of supposedly anonymized data was possible, and it has also been pointed out that PII has no technical meaning. Additionally, in some cases the manufacturer might want to release statistics either publicly or to an original equipment manufacturer (OEM). Given the challenges with properly de-identifying data, structured methods for performing de-identification should be used, rather than arbi-

trary removal of attributes believed to be sensitive.

A promising research area to help mitigate the re-identification problem is *differential privacy*, a privacy model that unlike most privacy models gives mathematically rigorous privacy guarantees. Although the research interest is large, the amount of real-world implementations is still small, since understanding differential privacy and being able to implement it correctly is not trivial. Therefore, in this position paper, we set out to answer the questions of how and when to use differential privacy in the automotive industry, in order to bridge the gap between theory and practice. Furthermore, we elaborate on the challenges of using differential privacy in the automotive industry, and conclude with our recommendations for moving forward.

3.1 Introduction

The ability to collect data from modern connected vehicles presents opportunities for increased analysis, which enables vehicle manufacturers to both improve existing as well as develop new services. For example, investigating driving behaviour would make it possible to learn more about the drivers' needs and preferences, allowing manufacturers to better cater to customers' needs. Especially, using machine learning on large data sets could result in interesting correlations that were previously unknown.

However, gathering data from vehicles is not only an opportunity for further analysis, but also a possible privacy risk to the individual drivers. A recent survey show that drivers' privacy concerns include disclosure of private information, car vehicle tracking and commercial use of their personal data [1]. Seeing as privacy is a concern for drivers when it comes to connected vehicles, the problem needs to be addressed by the manufacturers in order to maintain the drivers' trust. Moreover, the upcoming general data protection regulation (GDPR) [2] will soon enforce stricter handling of personally identifiable information (PII). Failure to comply with the GDPR may result in fines of up to

either €20,000,000 or 4% of the total worldwide annual turnover of the preceding financial year [2]. Even though the GDPR is a European law, it will affect all companies that sell vehicles to Europe, as this is where the data will be collected. It is therefore important that PII is handled with care in order to protect the company's brand, maintain the customers' trust as well as to meet the new legislation.

A common pitfall when de-identifying data is to only remove attributes than can obviously be classified as PII, such as VIN numbers. However, as pointed out by Narayanan and Shmatikov [3], defining and identifying PII is surprisingly difficult, and in fact, PII has no technical meaning. A vehicle has approximately 7700 unique signals [4], and while these signals may seem to be separate from PII, even observing a driver's behaviour for as short as 15 minutes is enough to fingerprint and identify a driver with high accuracy [5]. Furthermore, Gao et al. [6] showed that the driving speed in combination with an external road map is enough to trace the location of a vehicle with high accuracy, even though GPS data has been removed. In addition, Tockar [7] demonstrated that an "anonymized" version of NYC cab data, in combination with public data, contained enough information to track celebrities and identify passengers that visited sensitive locations in the city. Thus, all data should be treated as PII, since auxiliary data might be available to re-identify individuals. For example, the position of the car seat might not seem to be PII, but it is likely enough to distinguish between two drivers of the same car.

A promising privacy model with rigorous, mathematical privacy guarantees that could solve the previously mentioned problems is *differential privacy* [8, 9]. Intuitively, for an individual, the best privacy is achieved by not participating in a survey, as their data will not affect any statistics released from such a survey. Consequently, differential privacy aims to approximate one individual not being in the data set. Furthermore, differential privacy's privacy guarantees are robust and does not change over time, as it is *backward and forward proof*. That is, any current or future data set cannot affect the privacy guarantees offered by differential privacy.

As claimed by Dwork, differential privacy is able to provide high *utility*, accuracy, as well as high privacy in many cases [9]. This is a very desirable property, as there exists a trade-off between privacy and utility that is difficult to balance. Intuitively, this trade-off can be explained by investigating two extreme cases. Without utility, privacy makes little sense, as privacy without utility is satisfied when no data is gathered. However, full utility is achieved by publishing a raw data set, which does not give any privacy guarantees. As neither of these two cases are desirable, a trade-off between the two must be made.

While differential privacy shows promise, it can be challenging to use in real-world cases, as the utility is affected by different parameters. The most prominent real-world cases that use differential privacy have been presented by large companies, such as Apple [10] and Google [11], and only cover very limited use cases. In particular, for vehicular data, differential privacy has so far only been investigated for floating car data (FCD) [12]. Since differential privacy has not yet been established in the automotive domain, although there is a need for privacy-preserving analyses, we believe that differential privacy is a future trend that this paper will aid in paving the way forward for. Hence, the contribution of this position paper is a comprehensible introduction to differential privacy (Section 3.2, 3.3 and 3.4), where we investigate what type of differentially private analyses can be performed in the vehicular domain from a holistic perspective, not only for one specific data type. Furthermore, we provide recommendations (Section 3.5) for how to proceed when implementing differentially private analyses in the vehicle domain, and highlight the challenges (Section 3.6) involved with the implementation.

3.2 Differential Privacy

Differential privacy originates from statistical research and examples used often include queries on databases. It is important to note that differential privacy is designed to suit statistical queries that make predictions for large populations,

as it prevents inference of information about an entity. As has been pointed out, any meaningful privacy guarantees for differential privacy are not achievable when specific individuals in a data set should be identified [13]. For example, differential privacy will not return any useful information when we ask if Bob uses his company car on weekends.

The differential privacy definition, shown in Definition 1 [9], states that when the same query is run on two neighboring data sets, differing in at most one element, the difference between the probability of getting the same outcome of both queries is essentially negligible. In other words, the presence or absence of one single record does not affect the outcome of a query noticeably. Intuitively, the idea behind differential privacy is to produce a result to a statistical query that is *almost* indistinguishable whether or not one record is present or absent in the data set.

Definition 1 (ϵ-differential privacy)**.** *A randomized function $\mathcal{K}$ gives ϵ-differential privacy if for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq Range(\mathcal{K})$,*

$$Pr[\mathcal{K}(D_1) \in S] \leq exp(\epsilon) \times Pr[\mathcal{K}(D_2) \in S]$$

Two of the main properties of differential privacy are query *composability* and *post-processing* of data [14]. Being composable means that any results of differentially private analyses can be combined, in which case privacy degrades additively. Composability also allows several queries to target the same data. Other privacy models, such as *k*-anonymity [15], fails under composition [16], even with itself. Lastly, any post-processing conducted on data released under differential privacy can be included in any additional analyses, without increased risk to an entity [13].

The risk incurred on an individual is monitored by ϵ, which is sometimes also referred to as the *privacy guarantee*. When ϵ is set to a low value, it gives higher privacy at the cost of reduced utility, whereas a high ϵ gives lower privacy and higher utility. Thus, setting ϵ appropriately is a trade-off between utility and

privacy and should be carried out by an expert in the domain.

Another parameter involved is the privacy budget, which is a global parameter from which ϵ is deducted when a query is run. The privacy budget is being consumed by querying the database in order to maintain privacy, and the more queries the higher noise the answers receive. This response can intuitively be explained by an example including the game of twenty questions. In the game of twenty questions, the more questions that are answered, the closer the contestants get to the real answer. To counteract anyone from finding the real answer under differential privacy, the privacy budget enforces that each consecutive answer gets more vague. When the privacy budget is depleted, ϵ can only be set to zero, which means answers will no longer return any meaningful information about the data.

There are many different ways of achieving differential privacy, as any function K that fulfills Definition 1 is differentially private. The reason for why there are many different algorithms is that they are data dependent, and the utility from a differentially private algorithm changes depending on its input data [17]. Consequently, researchers are constantly inventing new algorithms that are optimized for their analysis, resulting in a vast number of differentially private algorithms with varying complexity and utility.

3.3 Release Mechanisms

The basic idea of a release mechanism, K from Definition 1, is to add probabilistic noise to the real query result. Different release mechanisms are better suited for different data types, such as numerical or categorical data. The lower bound of the accuracy of each release mechanism can also be proven mathematically in order to determine which mechanism is most likely to yield high utility.

Release mechanisms can also be deployed in two different modes: centralized

or local. In the centralized mode differential privacy is guaranteed by a trusted party, usually at the time when the database is queried. For local differential privacy on the other hand, each data point is collected under differential privacy in a distributed manner, meaning that noise is added locally. In this section we will describe the Laplace mechanism, the exponential mechanism and randomized response. Figure 3.1 shows an overview of the mechanisms and their respective characteristics.

Mechanism Name	Deployment Mode	Answer Data Type
Laplace Mechanism	Centralized (Off-board)	Numerical
Exponential Mechanism	Centralized (Off-board)	Categorical
Randomized Response	Local (On-board)	Categorical

Table 3.1: Comparison between the characteristics of three common differentially private mechanisms

3.3.1 The Laplace Mechanism

The Laplace mechanism, illustrated in Figure 3.1, works by adding controlled numerical noise drawn from a Laplace distribution to a query answer. To be able to hide changes in the data set, the query sensitivity, Δf, in combination with the privacy budget, ϵ, is used when generating the noise. The query sensitivity is the maximum impact removing or adding *any* record to the data set has on the query result.

Since the Laplace mechanism produces continuous numerical noise, it is suitable for queries that are also numerical. Queries can be either continuous or discrete, as differential privacy allows post-processing. In case of a discrete query, the output will be continuous, but can be rounded up to a discrete value without violating differential privacy.

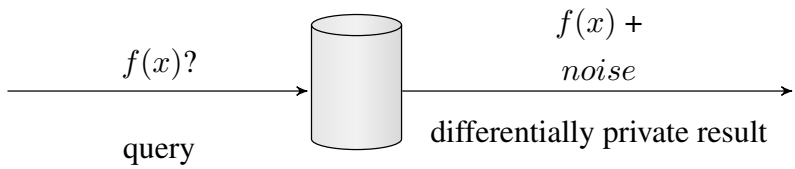


Figure 3.1: An illustration of a database with a Laplace mechanism that is used to release differentially private query answers

The Laplace mechanism is applied centrally by a trusted party. Thus, all raw data is kept in a database off-board, where each query result is released under differential privacy.

3.3.2 Exponential Mechanism

The exponential mechanism [18] is designed for categorical data, so the added noise is not numerical. Rather, the analyst provides a *utility function* that specifies the distance between the different categories. For example, the analyst might want to specify the distance between colors, where shades of the same color are closer than a different color. The exponential mechanism then uses the utility function to output a good answer to the query with higher probability than outputting an answer further from the truth. Thus, the exponential mechanism favors answers that have high utility for a given query input. Like the Laplace mechanism, the exponential mechanism is also applied centrally.

3.3.3 Randomized Response

Randomized response [19] was originally invented in 1965 to estimate the amount of people in the population that belong to a sensitive group. Since membership of the group is sensitive, the respondent has an incentive to lie if he or she is part of the group, which can cause a skewed distribution of answers. Therefore,

randomized response provides a protocol which gives the respondents *plausible deniability*, meaning that an analyst cannot tell if a given respondent lied or not while still being able to make predictions about the population.

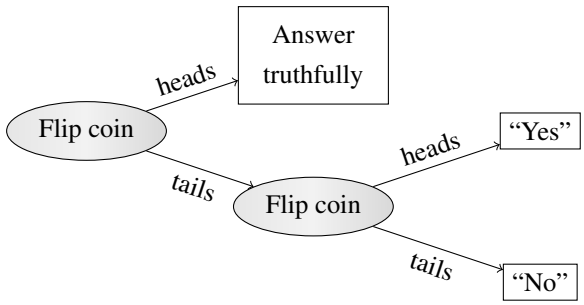


Figure 3.2: Randomized response, in this example following the protocol to answer the question "Do you text and drive?"

Randomized response enforces local differential privacy, and each driver follows the protocol in Figure 3.2 in order to respond under differential privacy. In order to interpret the results from randomized response, the analyst has to extract the number of people that where telling the truth using Bayes' theorem.

3.4 Privacy Guarantees

In order to utilize the privacy budget well, making it last longer than when using a naïve approach, privacy can be applied at event-level [20] rather than user-level. Event-level privacy protects a single event, such as a single data point where a driver is speeding, whereas user-level privacy typically protects an individual or an object such as a vehicle. The analyst defines what an event is, for example a reading of one single signal or something that happens after a certain condition is met. For example, one event might be that the airbag has been triggered, but it could also be one single reading of the engine temperature.

Essentially, the privacy level determines what or who should be protected by differential privacy, by determining what data points are considered to belong to one entity. In other words, if we choose user-level privacy for a car, all 7700 signals belong to that entity, whereas if we decide on event-level privacy, we can decide on a subset of those signals.

3.5 Advice

In theory, any query can be answered under differential privacy. In practice, however, some queries are better suited, since they offer a better trade-off between privacy and utility. Hence, in this section we will present some advice regarding how to proceed when creating a differentially private analysis for vehicular data.

3.5.1 Model the Domain

1) Decide the privacy level: Before starting to implement anything, it is important to define who or what privacy should be provided for. For example, if the driver's identity should be protected, user-level privacy needs to be used. Also, since a driver can drive more than one vehicle, this needs to be accounted for in the model.

In some cases, to improve the utility of the answer, the analyst might settle for only hiding certain events, such as speeding, in which case the analyst can choose to only provide privacy for the speed of the car. On the other hand, the analyst can also choose to hide only the time a driver was driving at a certain speed. In the case where only the time is hidden, the driver can deny that he or she was speeding since it is impossible to infer where the driver was driving. In other words, an analyst can choose to hide events of different sizes, such as only the time something happened or an entire driving pattern, and it is vital to define in advance what those events are.

Thus, modeling the kind of privacy that should be given and to whom needs to be done first, in order to decide the privacy level as well as finding a suitable value for ϵ.

3.5.2 Trusted Party or Not?

1) Decide deployment mode: The main advantage of local differential privacy is that each driver adds their own noise, as opposed to centralized differential privacy. Thus, local differential privacy, which can be implemented using randomized response, is carried out on-board whereas centralized differential privacy must be implemented off-board. Since randomized response is local, no trusted party is needed to gather all data, which also means companies never have to store or even get in contact with any sensitive data as it will be kept in the vehicle. Furthermore, on-board algorithms can also result in data minimization, meaning that less data is gathered from the driver, which is a property that is being promoted by the upcoming GDPR. However, the downside of local mechanisms is that achieving an adequate trade-off between privacy and utility is difficult in real-world cases [21].

3.5.3 Using the Privacy Budget

In order to get a good balance between utility and privacy, the privacy budget needs to be used with care. We believe there are certain techniques that could make the budget last longer, such as personalized budgets [22] (as opposed to a global budget) and random sampling. *1) Personalized budgets:* First, personalized budgets for differential privacy allows each record to keep its own budget, which means all records are not affected by queries that do not concern them. Using personalized budgets thus allows an analyst to keep the budget from being spent unnecessary, as he or she can query all vehicles of a certain model without also spending the budget for vehicles of other models.

From a data management perspective, another benefit of using personalized budgets is that even if there is no centrally controlled database gathering all the data, deductions to a global budget do not have to be communicated across databases as long as all data belonging to one entity remains in one database. Thus, a company can still keep several databases for different kinds of data without introducing dependencies between the databases.

2) Random sampling: Secondly, random sampling allows us to select a subset of records to query, and thus together with personalized budgets only spend the budget of that subset. Random sampling is especially appealing for big data sets, where a subset of the entire population still gives a good prediction. We believe that vehicular data fits this description.

3) Streaming data: Furthermore, we also believe the vehicle industry could benefit from enforcing differential privacy on streaming data instead of storing raw data in an off-board database, as all stored data would be sanitized. That is, vehicles could be selected to be part of a query, and then their replies could be released under differential privacy where the data is aggregated. In this way only the results from differentially private queries could be saved, and raw data thrown away. Since differential privacy offers post-processing, the data kept could then be used in any analysis. Apart from preserving privacy, this approach could also save storage space on the server side, and could also decrease the traffic used to upload data when queries only are issued on demand.

In the case of the streaming paradigm where vehicles are queried, each vehicle would have to keep track of its own budget and communicate it to the server, which would be possible when we use personalized budgets. Even though local differential privacy inherently is better suited for this setting, we believe this provides an alternative where local algorithms offer low utility.

3.5.4 Population Statistics, Never Individual Data

Differential privacy is designed to answer statistical queries that make predictions about the population, not for inferring information about individuals. Thus, if an analyst were to ask how often Bob uses the parking brake per week, the result would not be useful as the noise would likely be too high.

The accuracy of results can be vital if safety-critical functionality is to be developed from an analysis. In such cases, the upper-bound and lower-bound accuracy of a differentially private algorithm needs to be calculated before the analysis is carried out. If the differentially private algorithm does not provide a tight upper- and lower-bound on accuracy, the safety-critical functionality could be at risk by using data under differential privacy.

In these cases, there are two options: either the differentially private algorithm is modified (for example by rephrasing the query, see Section 3.5.5) to achieve higher accuracy, or the analysis is carried out without guaranteeing differential privacy on the company's own vehicles. For example, a case where differential privacy is not suitable is for function testing using high-resolution data from few vehicles.

3.5.5 Rephrase Queries

Rephrasing a query might result in better utility. *1) Target the population:* In some cases an inappropriate query, that targets individuals, can be rephrased into a query that targets the entire population. For example, if we want to find out when an engine is running outside of its specification, asking for in which vehicles this occurs would be a bad idea. On the other hand, what we are really interested in might not be which those cars are, but rather how many they are, to determine if it is common or not. In such a case it is possible to turn a bad query into a prediction about the population, a counting query in this case, which would provide a better answer to, approximately, the original query.

2) Change the query type: In other cases, the problem might not be that one individual is targeted, but that the query itself is prone to result in high noise. As an example, instead of asking for the average speed, the speed can be investigated from a histogram from which heavy-hitters can be identified. In other words, when the query sensitivity is high, transforming the query into a less noisy one is advisable, unless the difference between the query result and the proportional noise is small.

3.5.6 Dealing with Query Sensitivity

One issue with query sensitivity is that in practice it can be hard to define. Therefore, in some cases, the query sensitivity needs to be set to the physical maximum of a parameter, which is unlikely but necessary.

1) Query a large data set: Some queries, such as sums and averages, tend to have high query sensitivity. For vehicles, the analyst might then when defining the query sensitivity refer to the maximum value that can be held in a certain register in the vehicle. While these queries can still be used, the noise will be easier to hide when a larger data set is queried. Thus, the data set's size is more important in cases where the query sensitivity is high rather than in cases where it is constant, such as counting queries and histograms.

2) Fixed sensitivity through cropped ranges: The way we suggest for dealing with high query sensitivity is to crop the ranges and set a fixed max and min value. All values outside of range must not be used in the analysis, as they would not be protected in this case. The chosen range itself also leaks information about what range is expected to be normal. When the range itself is sensitive data, the range must be decided under differential privacy.

However, if the range is not well-known, it is possible to accidentally set the range to an interval which a large part of the values fall outside of. To be able tweak an incorrectly set range in a differentially private manner, we suggest

creating one bin on each side of the range that catches all outside values. When the side-bins are fuller than a certain threshold, it indicates a problem with the chosen range, which then needs to be redefined.

3.5.7 Applicable Analyses

1) Histograms and counting queries: Histograms and counting queries are particularly suited for the Laplace mechanism, as pointed out by Dwork [23]. The reason for this is that histograms and counting queries have a fixed sensitivity, which generally results in low noise that is independent of the data set's size. Consequently, when the data set queried is small, histogram and counting queries are especially appropriate.

2) Numerical queries: Any other numerical query is also possible to implement under differential privacy using the Laplace mechanism. However, the Laplace mechanism is highly dependent on the type of query being asked, as each query type has its own sensitivity, Δf. For example, if we want to calculate the average speed of a vehicle, we need to account for the largest possible change adding or removing any data point to the set can have on the average. Consequently, we must assume the worst case, which in this case is adding the highest possible speed to the data set. Thus, the sensitivity is the difference between the maximum and minimum speed *possible*. The sensitivity will then affect the proportion of noise that is added to the query result, and thus we suggest choosing a query which has lower sensitivity as it generally will yield lower noise than a high sensitivity query.

3) Categorical queries: For data where adding noise makes little sense, such as categorical data, the exponential mechanism can be used. One such example is when asking for the most popular car colors, as adding numerical noise to colors does not make sense. Another example would be if we want to find out what button on the dashboard is pushed the most times.

3.6 Challenges

There are many challenges with properly implementing a differentially private analysis in real-world cases. In this section we point out some of the most prominent ones for vehicular data.

3.6.1 Setting the Privacy Budget

To reason about ϵ, the domain must first be modeled in such a way that the entity to protect has been defined through setting the privacy level. ϵ is then the factor of indistinguishability between any two entities. Consequently, setting ϵ to a meaningful value is difficult, as ϵ is a relative measure of privacy risk [24]. In other words, the appropriate value of ϵ is affected by the type of data being released. Thus, the risk of two differentially private algorithms cannot be compared by their value of ϵ. This problem is not unique to vehicular data, but follows inherently from the definition of differential privacy.

While how to choose ϵ appropriately remains an open research question, Lee and Clifton as well as Hsu et al. propose practical solutions to the problem. Lee and Clifton suggests choosing ϵ based on the individual's risk of being re-identified [24], whereas Hsu et al. [25] propose that ϵ should be chosen based on an economic model. While no approach is clearly better than the other, both solutions provide an interpretation of what the privacy guarantees mean to a participant, making it possible to communicate the risk accordingly.

3.6.2 Multidimensional Time Series Data

Compared to other systems, preserving the privacy of vehicles is particularly difficult since they are highly complex systems that generates vast amounts of data from thousands of signals. To make matters worse, vehicle signals can be gathered continuously over time. Consequently, as the amount of available

data simplifies identifying a particular vehicle, hiding the presence of a specific vehicle in the data set becomes more difficult than hiding fewer connected data points.

Because of the multidimensional time series nature of the data, performing more than one analysis with high utility that guarantees user-level privacy becomes infeasible. User-level privacy would also not allow the analyst to reset the budget, not even after years of using the same budget. Consequently, we believe that in order to maintain utility, analyses can only provide event-level privacy.

On a positive note, providing event-level privacy can save the manufacturer the trouble of maintaining the privacy budget between different systems, as it results in separate privacy budgets for each system.

An open issue that we need to solve in this area is interpreting what event-level differential privacy means for a driver, as it is an individual that ultimately wants the privacy. For example, what does it mean from a privacy perspective if we only hide at what point in time the battery had a certain temperature? Event-level privacy might be more feasible than user-level privacy from a utility perspective, but every case must be investigated to make sure the privacy guarantees in such a situation makes sense to an individual as well.

3.7 Conclusion

For vehicular data, differential privacy can be especially tricky to enforce due to the fact that vehicles contain a system of thousands of dependent signals collected over time. Consequently, the automotive domain is very complex from a privacy perspective. However, as differential privacy is the only privacy model that provides provable privacy guarantees, this is currently the only robust way of mitigating re-identification attacks on data while maintaining utility. Thus, we believe that the automotive industry will benefit from carrying out

their privacy-preserving analyses under differential privacy.

In order to properly implement differential privacy, it is vital that the company first model the privacy within their domain, to determine what they are trying to protect. From the model, the company can then define what signals an event should consist of, and the model also makes it possible to reason about a suitable value for ϵ. Only after the modeling has been done can the implementation details of the analysis be decided.

Differential privacy should be used to answer statistical questions about a population. Since differential privacy aims to protect the privacy of each entity, it is not suitable for detecting anomalies. Because of this, analyses on high-resolution data from few vehicles, such as when performing function testing, should not be carried out under differential privacy. Any other statistical queries can be answered under differential privacy, but we believe that one of the main problems with introducing differential privacy in the automotive domain is maintaining high utility for the analyses. Thus, we have investigated ways of being able to spend the privacy budget wisely.

We believe that in order to enforce differential privacy for vehicular data in a sustainable way, personalized budgets, random sampling as well as event-level privacy are key to high utility. Rephrasing queries as well as cropping ranges of queries is also something that can make differential privacy more applicable. Furthermore, we believe that by issuing queries to vehicles on the go using the streaming paradigm or local differential privacy, there is potential to save both storage space and bandwidth while preserving privacy at the same time.

In the end, we believe differential privacy shows promise for the vehicle industry. However, more work still needs to be put into interpreting the meaning of ϵ as well as event-level privacy from a customer's perspective, as the meaning will differ on a case-by-case basis.

Bibliography

[1] FEDERATION INTERNATIONALE DE L'AUTOMOBILE (FIA) REGION I. *What Europeans Think about Connected Cars*. Jan. 29, 2016. URL: http://www.mycarmydata.eu/wp-content/themes/shalashaska/assets/docs/FIA_survey_2016.pdf (visited on 01/24/2017).

[2] European Parliament, Council of the European Union. *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)*. May 4, 2016. URL: http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679%5C&qid=1473158599287%5C%5C%5C&from=EN (visited on 09/06/2016).

[3] A. Narayanan and V. Shmatikov. "Myths and Fallacies of "Personally Identifiable Information"". In: *Commun. ACM* 53.6 (June 2010), pp. 24–26. (Visited on 01/27/2017).

[4] P. Kleberger et al. "Towards Designing Secure In-Vehicle Network Architectures Using Community Detection Algorithms". In: *2014 IEEE Vehicular Networking Conference (VNC)*. Dec. 2014, pp. 69–76.

[5] M. Enev et al. “Automobile Driver Fingerprinting”. In: *Proceedings on Privacy Enhancing Technologies* 2016.1 (2015), pp. 34–50.

[6] X. Gao et al. “Elastic Pathing: Your Speed Is Enough to Track You”. In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. UbiComp ’14. New York, NY, USA: ACM, 2014, pp. 975–986.

[7] A. Tockar. *Riding with the Stars: Passenger Privacy in the NYC Taxicab Dataset*. neustar // Research. Sept. 15, 2014. URL: https://research.neustar.biz/2014/09/15/riding-with-the-stars-passenger-privacy-in-the-nyc-taxicab-dataset/ (visited on 02/15/2017).

[8] C. Dwork et al. “Calibrating Noise to Sensitivity in Private Data Analysis”. In: *Theory of Cryptography*. Ed. by S. Halevi and T. Rabin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 265–284.

[9] C. Dwork. “Differential Privacy”. en. In: *Automata, Languages and Programming*. Ed. by M. Bugliesi et al. Lecture Notes in Computer Science 4052. Springer Berlin Heidelberg, Jan. 2006, pp. 1–12.

[10] A. Greenberg. *Apple’s ‘Differential Privacy’ Is About Collecting Your Data — But Not Your Data*. WIRED. June 13, 2016. URL: https://www.wired.com/2016/06/apples-differential-privacy-collecting-data/ (visited on 01/30/2017).

[11] Ú. Erlingsson et al. “RAPPOR: Randomized Aggregatable Privacy -Preserving Ordinal Response”. In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. CCS ’14. New York, NY, USA: ACM, 2014, pp. 1054–1067.

[12] F. Kargl et al. “Differential Privacy in Intelligent Transportation Systems”. In: *Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks*. WiSec ’13. New York, NY, USA: ACM, 2013, pp. 107–112.

[13] S. Vadhan. “The Complexity of Differential Privacy”. 2016. URL: http://privacytools.seas.harvard.edu/files/

privacytools/files/complexityprivacy_1.pdf (visited on 02/06/2017).

[14] C. Dwork et al. "Exposed! A Survey of Attacks on Private Data". en. In: *Annual Review of Statistics and Its Application* 4.1 (Mar. 2017), pp. 61–84.

[15] P. Samarati. "Protecting Respondents Identities in Microdata Release". In: *IEEE transactions on Knowledge and Data Engineering* 13.6 (2001), pp. 1010–1027.

[16] S. R. Ganta et al. "Composition attacks and auxiliary information in data privacy". In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, Aug. 24, 2008, pp. 265–273.

[17] M. Hay et al. "Principled Evaluation of Differentially Private Algorithms Using DPBench". In: *Proceedings of the 2016 International Conference on Management of Data*. SIGMOD '16. New York, NY, USA: ACM, 2016, pp. 139–154.

[18] F. McSherry and K. Talwar. "Mechanism Design via Differential Privacy". In: *48th Annual IEEE Symposium on Foundations of Computer Science, 2007. FOCS '07*. 48th Annual IEEE Symposium on Foundations of Computer Science, 2007. FOCS '07. Oct. 2007, pp. 94–103.

[19] S. L. Warner. "Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias". In: *Journal of the American Statistical Association* 60.309 (Mar. 1965), pp. 63–69.

[20] C. Dwork et al. "Differential privacy under continual observation". In: Proceedings of the forty-second ACM symposium on Theory of computing. ACM, 2010, pp. 715–724. (Visited on 07/12/2016).

[21] R. Chen et al. "Private spatial data aggregation in the local setting". In: *Data Engineering (ICDE), 2016 IEEE 32nd International Conference on*. IEEE, 2016, pp. 289–300.

[22] H. Ebadi et al. "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 Press, 2015, pp. 69–81.

[23] C. Dwork. "Differential Privacy: A Survey of Results". In: *Theory and Applications of Models of Computation*. Ed. by M. Agrawal et al. Lecture Notes in Computer Science 4978. Springer Berlin Heidelberg, Jan. 1, 2008, pp. 1–19.

[24] J. Lee and C. Clifton. "How Much Is Enough? Choosing ε for Differential Privacy". en. In: *Information Security*. Ed. by X. Lai et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2011, pp. 325–340.

[25] J. Hsu et al. "Differential Privacy: An Economic Method for Choosing Epsilon". In: *2014 IEEE 27th Computer Security Foundations Symposium*. July 2014, pp. 398–410.

Paper III

Mathias Johanson, Jonas Jalminger, Emmanuel Frécon, **Boel Nelson**, Tomas Olovsson, Mats Gjertz

Joint Subjective and Objective Data Capture and Analytics for Automotive Applications

2017 IEEE 86th Vehicular Technology Conference (VTC-Fall), Toronto, ON, 2017, pp. 1-5, doi: 10.1109/VTCFall.2017.8288366.

Presented at *2nd IEEE International Workshop on Vehicular Information Services for the Internet of Things (VISIT 17)*

Toronto, Canada

September 24, 2017

Joint Subjective and Objective Data Capture and Analytics for Automotive Applications

Abstract

In this paper we describe a novel technological framework for capture and analysis of both objective measurement data and subjective user experience data for automotive applications. We also investigate how the framework can be extended to address privacy issues by enforcing a rigorous privacy model called differential privacy. The system under development integrates a telematics system with a smartphone app service architecture and a data-driven analytics framework. The hypothesis is that the framework will improve the opportunities of conducting large scale user trials of automotive functions and services, while improving the quality of collected data. To achieve this, a number of challenges are addressed in the paper, including how to design the subjective data capture mechanisms to be both simple to use yet powerful, how to correlate subjective data with objective measurement data, and how to protect the privacy of users.

4.1 Introduction

A key to competitiveness in the automotive industry is to be highly responsive to customer needs and expectations as well as market trends. One way to achieve this is to collect and analyze data from connected vehicles to find out how the customers use the product and how the product performs in different situations. The opportunities to employ data capture and analytics for knowledge-driven product development, whereby engineering and design decisions are made based on hard facts rather than best practices and tacit knowledge is gaining strong momentum in the automotive industry [1]. Sophisticated telematics systems and cloud-based analytics frameworks are emerging for these types of applications [2], but what is generally missing is a good way to couple the collected vehicular data and usage data to customer experience data. How the vehicle and user behaves is only one side of the story, the other being how the user experiences the product or would like to experience the product. The objective data being collected through telematics services therefore need to be complemented with subjective data about the customers' experiences of using the product.

The traditional approach to subjective data collection in the automotive industry is through surveys based on questionnaires and interviews with selected customers. However, this type of subjective data collection is time consuming and the collected data sets are typically difficult to correlate with objective measurement data. What the customer says about the way he or she uses a product does not necessarily correspond to how he or she actually uses the product, nor with how the user would like to use the product or what new features and services are desired. Furthermore, subjective data quality is commonly low since there is a considerable separation in time and space between actually using a product and responding to a traditional questionnaire. The experience the user had while using the product is easily dimmed, forgotten or altered by the passing of time and change of locale. Moreover, when it comes to advanced active safety and autonomous driving services, the volume and complexity of data that need to

be collected is high, so a scalable architecture with a high level of automation is needed for capture and analysis of data.

To overcome these problems, we suggest an approach based on a technological framework for coordinated capture and analysis of both objective and subjective data — the latter through the use of a smartphone app which can present tailored questions to selected users to capture specific information about particular events triggered by conditions detected in each user's vehicle during usage. The subjective data submitted through the app is uploaded to a cloud-based analytics framework where objective data, collected from in-vehicle measurement systems are also available for combined analytics. Since the collected data might be privacy sensitive to users, we also investigate how the data can be collected in a privacy-preserving way. This gives the opportunity to carry out large-scale collection of data and automated data-driven analysis, with much higher information quality and in shorter time compared to traditional approaches, reducing the time to market for new product features and customized service offerings. The concept is illustrated in Figure 4.1.

4.1.1 Target Applications

To explore the opportunities of joint subjective and objective data collection, we have developed a proof-of-concept system targeting primarily active safety applications, but with a great potential to be used for many other automotive applications where subjective user data is important, including climate comfort, noise-vibration-harshness (NVH) and ergonomics.

Since active safety and autonomous driving functions increasingly rely on machine learning algorithms that typically require large volumes of training data, systems that can facilitate the collection of relevant training data sets are very important. Finding the relevant training data sets typically requires human intervention, e.g. to tag or classify whether a given situation belongs to a given category. With the user in the loop through the smartphone app, our approach

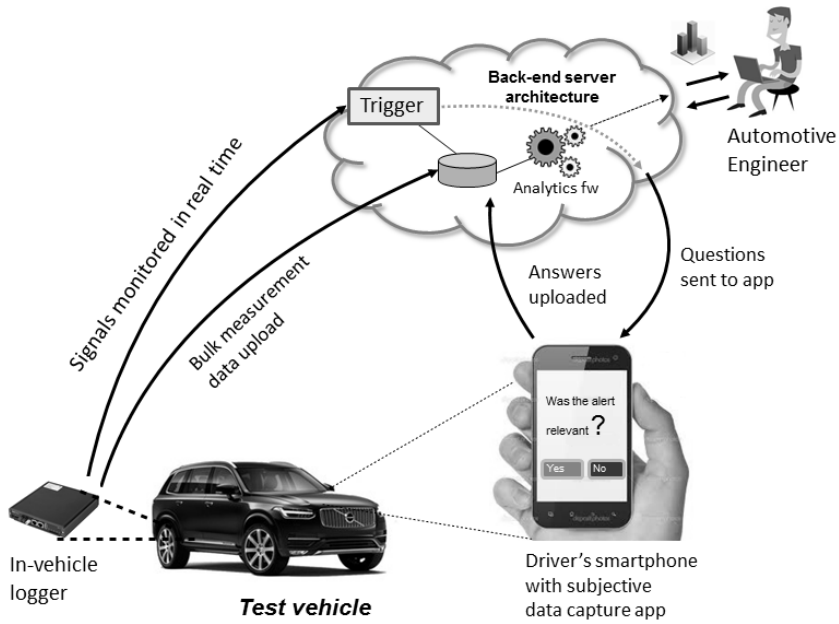


Figure 4.1: Joint subjective/objective data capture and analytics concept

gives tremendous opportunities to build up a large database of training data. Since the sensor data input to active safety systems typically include complex data types such as video and radar/lidar images, this also affects the design and configuration of the onboard logging devices and corresponding telematics services used to capture and communicate measurement data.

4.2 Challenges

In the design of the technological framework we have identified a number of challenges that need to be addressed. Some of the more demanding challenges are:

- ▷ How can we design the subjective data capture app in a way that makes

it easy and safe to use in a vehicle, even while driving?

- ▷ How can we design a triggering mechanism to decide when a particular question or set of questions should be posed to a particular user? The triggering mechanism must be versatile and flexible to be usable for all relevant use cases.
- ▷ How can we cater for follow-up questions that depend on answers to previous questions?
- ▷ How can we protect the privacy of users while at the same time providing automotive engineers with as powerful data collection and analytics tools as possible?

Each of the listed challenges are discussed in the text in the upcoming sections.

4.3 A Framework for Joint Subjective-Objective Data Capture and Analytics

The proof-of-concept framework is composed of the following main components:

+ An in-vehicle data capture and telematics system, making it possible to monitor and transmit in-vehicle (CAN bus) signals,
+ A cloud-based server infrastructure, including database storage, web-based user interface front-end, and application programming interfaces to provide controlled access to the information resources and framework services,
+ A smartphone app to which questions to vehicle users can be pushed from the server infrastructure, and answers recorded and uploaded to the database,
+ An analytics service architecture, enabling automated data-driven analysis of data originating from connected vehicles and smartphone apps,

+ A app questionnaire authoring tool for designing the questions to be sent to users of the smartphone app,
+ A concept for a privacy-preserving framework based on differential privacy.

An overview of the software architecture of the system is shown in Figure 4.2.

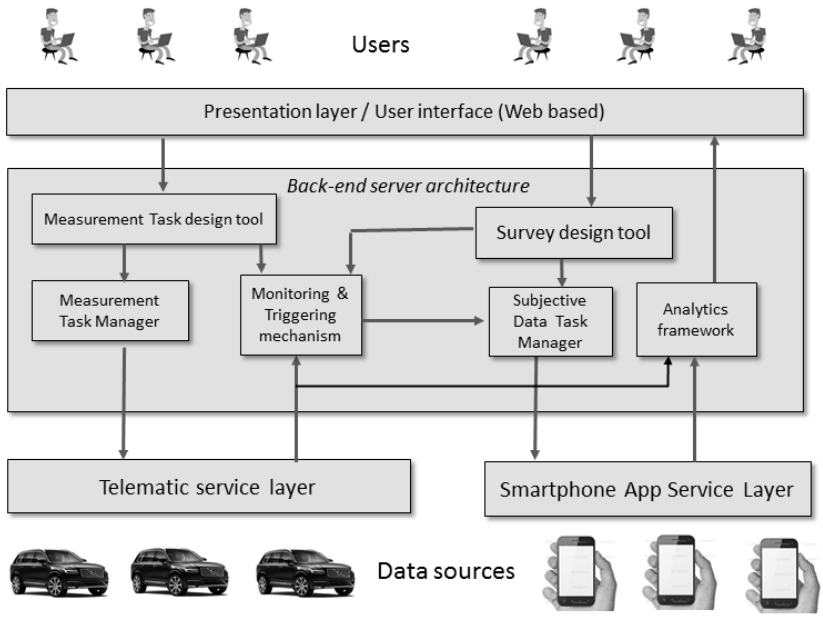


Figure 4.2: Software Architecture of the framework for joint subjective/objective data capture and analytics

4.3.1 Telematics System

The core component of the telematics system (called WICE) is a Linux-based data capture and communication unit installed in vehicles. The unit executes measurement tasks that support data capture both by passive in-vehicle communication bus monitoring and active diagnostic services. The captured data

is uploaded to a cloud-based server infrastructure using 2G, 3G or 4G wireless mobile data communication. The telematics unit also provides additional services such as GPS-based positioning and fleet management.

The communication architecture can handle both bulk upload of data and real-time streaming of data without the need to store it on the solid state disks of the telematics units. For most data capture services, measurement data is stored to disk while a data logging task is running, and then pre-processed and uploaded at the end of the vehicle's driving cycle (i.e at ignition-off). The streaming mode is used for time-sensitive applications, such as positioning services where it is important to show the current location of moving vehicles.

4.3.2 Smartphone App and App Service Architecture

The Smartphone App (see Figure 4.3) is implemented on top of the Ionic Framework [3] in order to target the most common mobile ecosystems from a single code base. This was deemed necessary in order to rapidly iterate the design throughout the life of the project. Ionic is one of the frameworks making it possible to use regular Web technologies (JavaScript, HTML, CSS, etc.) to develop native-looking apps. A number of specific libraries allow access to local hardware in a manner that hides most of the differences between iOS and Android. There are three major functions provided by the app:

+ Registering cars to app user accounts. Car registration is made through manual entry of the car's unique VIN, or through scanning a barcode representing this same identifier and usually printed onto the car's front window.
+ Each user account carries a few profile details in order to be able to target specific driver details: persons of above/below average height, in specific age categories, etc.
+ Receive and respond to "polls" in order to collect subjective information whenever the back-end has discovered a set of matching metrics that require complementary details for a deeper understanding.

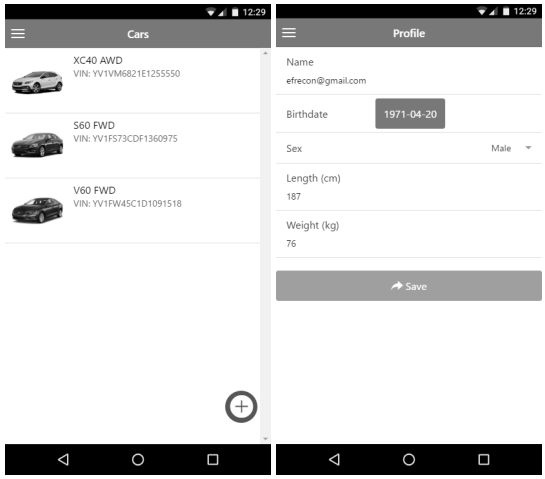


Figure 4.3: The screenshot to the left shows the landing page of the smartphone app, listing all cars that are registered to a given account. Note the "hamburger" menu in the top-left corner to access the rest of the app's functions, and the "+" floating button to trigger car registration. The screenshot to the right shows the profile screen aimed at collecting anthropomorphic data.

Care has been taken to minimize the size and intrusion of the polls as much as possible. Most polls will only contain a few questions, and questions can be conditional, i.e. only asked depending on previous answers within the same poll. The app accepts remote polls even under driving circumstances. However, polls are then read out loud using the mobile platform specific Text-to-Speech (TTS) functions and speech recognition is used to collect answers. Whenever alternatives are offered, small meaningful pieces of these sentences can be used to acknowledge the specific alternative. TTS is also used to inform about errors and progression, so as to engage the driver in a hands-free dialog. All questions and polls are also present on the smartphone screen, making it possible to answer using touch if necessary or preferred (see Figure 4.4). The UI uses large, clean and colour-coded buttons to facilitate interaction in all situations, including a bumpy road.

Sometimes it is desirable for polls to be sent in several steps. For example, first as soon as a telematics function has triggered (in order to capture the driver's answer in the heat of the action), but also later once the car has come to a stop (in order to capture further details about the specific event). These chains are not handled by the polls themselves, but rather through information exchange with the back-end system. Typically, two (or more) polls will be sent by the back-end, possibly conditionally, to capture these situations appropriately. However, the current implementation of the app collects phone position data to approximate speed, and in order to be able to cover these cases without back-end intervention, should that turn out to be necessary in future versions.

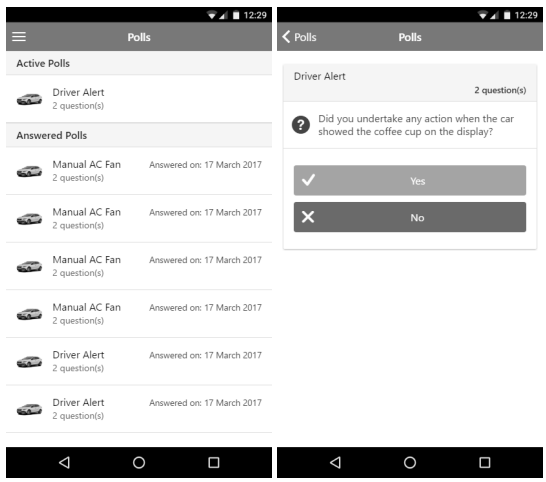


Figure 4.4: The screenshot to the left shows the list of polls as seen from the app. Upon answer, polls automatically get sorted into a separate list, shown at the bottom of the screen in order to provide some progress and history feedback. The screenshot to the right shows a typical yes/no question from a poll; the app also supports more elaborate questions with several alternatives. The "coffee cup" is a direct reference to how fatigue alerts are mediated to drivers in the car.

Several drivers/passengers can declare ownership of a single car. At present, relevant polls are sent to all registered users. However, this could be alleviated

through automatically detecting which of the registered users is currently (or has just) visited the car. We intend future versions of the app to communicate with the car's infotainment system as the main source of this type of information. This will also bring the possibility to offer an option that automatically turns on speech recognition (and TTS) when the phone is used in the car. This would prevent polls to be read out loud once the driver has stepped out of the car (which might be disturbing or embarrassing).

4.3.3 Back-end Server Architecture and Analytics Framework

The back-end architecture consists of two frameworks. One is the subjective data capture framework described in this paper which handles the polls and the other is the telematics and analytics framework called WICE [2] which delivers the signals from the car to the back-end data processing framework and provides data processing functions to analyze and visualize the captured data. In order for polls to be delivered to users the person creating the questionnaire must decide upon which set of vehicles should receive the poll when a certain condition occurs and this is done through a web-based tool for creating and managing polls.

The following takes place when a vehicle delivers data to the back-end.

1. In-vehicle signals are streamed in real time by the telematics system from connected vehicles to the back-end processing framework. Which signals are streamed is defined in a pre-configured measurement set-up.
2. Configurable trigger conditions are evaluated to find whether an event that is of interest has occurred. The trigger conditions are boolean expressions involving signals being streamed, for example `VehicleSpeed >50 AND Gear=3`. When a trigger condition specified for a specific poll evaluates to true, a service is called which sends the poll to the app which has been registered for the vehicle originating the data stream wherein the interesting event occurred.

3. Once the user has answered the poll, the answer is uploaded to the back-end framework and stored in a database, for subsequent analytical processing.

In some cases it is desirable that follow-up questions are posed when the user has responded in a specific fashion. Therefore the back-end framework must be able to evaluate trigger conditions that also include answers to previous polls in order to able to trigger follow-up polls.

The analytics framework, which is under development, is based on a data-driven approach, whereby data sets uploaded from connected vehicles and apps are automatically analyzed. Analysis result are stored in a knowledge base and made available for visualization, typically as histograms, pie charts or similar.

4.4 Case Studies and User Trials

The technological framework under development will be tested and evaluated in a case study at Volvo Cars wherein two different active safety features are focused: Driver Alert Control (DAC) and Forward Collision Warning (FCW). The DAC system is a driver fatigue detection and warning system. Subjective data is in this case collected to verify whether drivers actually are tired when the DAC system triggers, and to follow up whether they take a break as the system suggests. The FCW system alerts the driver when there is risk for a collision. Subjective data is collected to verify whether issued collision warnings are relevant. The purpose of the case study is to collect subjective user experience data from field trials and to analyze the data together with (objective) measurement data in order to improve the DAC and FCW systems. The hypothesis is that the technological framework presented in this paper will facilitate the orchestration of this kind of user experience surveys with a potentially large number of participating users, and to improve the quality of the data being collected.

4.5 Privacy Issues

While our approach to collect user data opens up new opportunities for improved, data-driven analytics, it also has privacy implications for the drivers that need to be addressed. For example, if a driver has a high number of FCW, it can indicate that the driver is reckless or aggressive, as he or she is often about to collide with objects. An additional privacy issue in this particular setting is that follow-up questions can be issued based on previous answers, which makes the fact that the follow-up question is sent reveal sensitive information. As an example, if a driver ignores the DAC even though he or she is tired, and confesses that this is the case through submitting subjective data during a follow-up poll, this information could be incriminating if the driver is later involved in a traffic accident.

Traditionally, analysts would choose to *de-identify* data, often through removing certain identifiers, such as the vehicle identification number (VIN) and the license plate from the data set. However, real-world examples [4, 5] has shown that de-identification often fails, allowing individuals to be *re-identified*. Examples from the automotive domain where re-identification has been possible include deducing the location of a car based on its speed [6] and fingerprinting drivers from their driving style [7].

In order to protect the driver's privacy, we suggest that data is gathered under *differential privacy*. Differential privacy [8] gives mathematically proven, robust privacy guarantees, which is not provided by any other privacy model. Definition 2 shows the formal definition of differential privacy [9]. Intuitively, differential privacy aims to simulate the best privacy for an individual: namely when he or she has opted out of the analysis. Essentially, differential privacy provides privacy by introducing some inaccuracy, noise, to a real answer. The privacy risk to an individual is then monitored by a privacy budget, which is usually shared by all participants.

Definition 2 (ϵ-differential privacy)**.** *A randomized function $\mathcal{K}$ gives ϵ-differential privacy if for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq Range(\mathcal{K})$,*

$$Pr[\mathcal{K}(D_1) \in S] \leq exp(\epsilon) \times Pr[\mathcal{K}(D_2) \in S]$$

To address the privacy issues of the smartphone app, we suggest that a framework[iv] for personalized local differential privacy (PLDP) based on randomized response [10] is developed and used when issuing questions from the app. Randomized response is a surveying technique that was invented to avoid evasive answers, for example by lying, from respondents. Randomized response is implemented by letting the respondent flip a coin to determine whether to lie or to answer truthfully, and if the respondent is asked to lie, he or she again flips a coin to determine what to answer. As the one collecting the answer does not know whether the respondent tells the truth or provides the random answer determined by the coin, randomized response is said to give *plausible deniability*. When the analyst wants to perform an analysis on the data, he or she uses Bayes' theorem in order to extract the truthful answers. This way data can be collected without it being possible trace a reply back to a specific individual, and also giving the respondents an incentive not to lie unless the coin tells them to.

To address privacy in our architecture, the PLDP framework would be placed in a privacy preservation layer above the smartphone app service layer, and work as an application programming interface (API) used for the questions in the app. Previously, PLDP has only been investigated theoretically [11], and practical implementations do not yet exist. The updated version of the software architecture is shown in Figure 4.5.

Similarly, data from the telematic service layer should also be passed through a privacy preservation layer. The main challenge here is to be able to ask follow-

[iv] The use of PLDP in this context is ongoing joint work with Hamid Ebadi and David Sands at Chalmers University of Technology

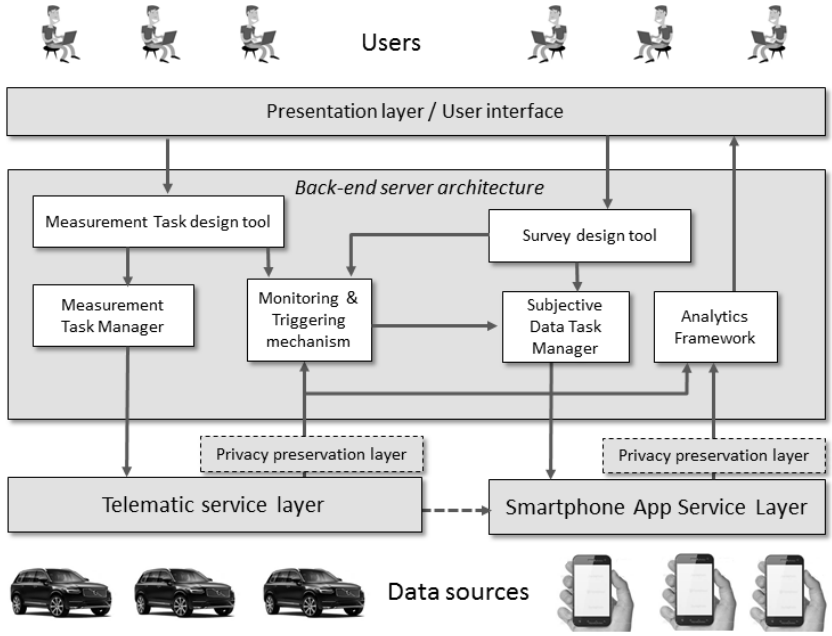


Figure 4.5: Updated software architecture of the framework with privacy in mind

up questions, without letting the back-end server learn the answer to the original questions. Therefore, the polls cannot be issued by the back-end server, but instead will be sent by the telematics server layer as it has access to the car's data. Then, the back-end server chooses a number of cars, uniformly at random, to answer a question. In this way, answers will only be uploaded once the back-end server has chosen that car to participate in a question.

The main implications of a PLDP framework for cars are:

- Local differential privacy does not require a trusted party, as privacy is enforced before the answer is sent to the back-end server. No sensitive data will therefore be uploaded.

- Local differential privacy also gives the driver an incentive not to lie, as raw data never reaches the back-end server.
- Personalized budgets allow for more flexible budgeting than traditional, global budgets, thus allowing for more questions being answered with high accuracy than when using global budgets.
- For each user, a privacy budget needs to be stored and managed, as budgets are personalized.
- Answers to polls need to be saved, in a private state, in the smartphone app.

4.6 Conclusions and Future Directions

In this paper we have explored the opportunities and challenges of joint subjective/objective data capture and analytics for automotive applications. Access to subjective data and sophisticated analytics frameworks in the testing, verification and validation phases of product development promises improved product quality and shorter development cycles, reducing the time to market for new products. We believe that the framework presented in this paper contributes strongly to this. Our future work includes integration of more advanced analytics and visualization mechanisms into the framework and to improve the overall design based on experiences from the case study described in section 4.4. Furthermore, we have also investigated how to extend the data capture to collect both the subjective user data and the objective car data in a privacy-preserving fashion under differential privacy.

4.6 Acknowledgement

This work was co-funded by VINNOVA, the Swedish Governmental Agency for Innovation Systems.

Bibliography

[1] R. Walker. *From Big Data to Big Profits: Success with Data and Analytics*. en. Oxford University Press, July 2015.

[2] M. Johanson et al. "Big Automotive Data: Leveraging large volumes of data for knowledge-driven product development". English. In: *Big Data (Big Data), 2014 IEEE International Conference on*. IEEE, Oct. 2014, pp. 736–741.

[3] Drifty. *Ionic Framework*. URL: `https://ionicframework.com/` (visited on 03/28/2017).

[4] P. Samarati and L. Sweeney. *Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression*. Tech. rep. SRI International, 1998.

[5] A. Narayanan and V. Shmatikov. "Robust De-anonymization of Large Sparse Datasets". In: *IEEE Symposium on Security and Privacy, 2008. SP 2008*. May 2008, pp. 111–125.

[6] X. Gao et al. "Elastic Pathing: Your Speed Is Enough to Track You". In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. UbiComp '14. New York, NY, USA: ACM, 2014, pp. 975–986.

[7] M. Enev et al. "Automobile Driver Fingerprinting". In: *Proceedings on Privacy Enhancing Technologies* 2016.1 (2015), pp. 34–50.

[8] C. Dwork et al. “Calibrating Noise to Sensitivity in Private Data Analysis”. In: *Theory of Cryptography*. Ed. by S. Halevi and T. Rabin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 265–284.

[9] C. Dwork. “Differential Privacy”. en. In: *Automata, Languages and Programming*. Ed. by M. Bugliesi et al. Lecture Notes in Computer Science 4052. Springer Berlin Heidelberg, Jan. 2006, pp. 1–12.

[10] S. L. Warner. “Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias”. In: *Journal of the American Statistical Association* 60.309 (Mar. 1965), pp. 63–69.

[11] R. Chen et al. “Private spatial data aggregation in the local setting”. In: *Data Engineering (ICDE), 2016 IEEE 32nd International Conference on*. IEEE, 2016, pp. 289–300.

Paper IV

Boel Nelson

Randori: Local Differential Privacy for All

Under submission

Randori: Local Differential Privacy for All

Abstract

Polls are a common way of collecting data, including product reviews and feedback forms. We propose giving quantifiable privacy guarantees through the statistical notion of *differential privacy* for polls. Still, since polls can contain of follow-up data that is inherently dependent data, implementing differential privacy correctly may not be straight forward in this setting. Moreover, apart from successfully implementing differential privacy, the inherent trade-off between accuracy and privacy also needs to be balanced.

Motivated by the lack of tools for setting the privacy parameter (ε) and need for correct implementations [1], we present RANDORI, a set of novel open source tools for differentially private poll data collection. RANDORI is designed to offer both privacy guarantees and accuracy predictions for *local differential privacy*. Hence, our tools provide analytical predictions of the accuracy of a poll that can be taken into account when setting ε. Furthermore, we show that differential privacy alone is not enough to achieve end-to-end privacy in an interactive server-client setting. Consequently, we also investigate and mitigate implicit data leaks in RANDORI.

5.1 Introduction

Polls are a widely used way of collecting data. For example, one is often asked to fill out a review after purchasing an item online. Now, these polls can consist of an arbitrary number of intertwined questions. For example, after purchasing an item online, one might be asked *"How do you feel about your purchase?"* with answer alternatives '*Happy*', '*Neutral*' and '*Unhappy*'. Next, a merchant can also choose to add a follow-up question asking *"What's the reason you feel unhappy?"* to all respondents that answer that they were unhappy with their purchase. Consequently, a poll's structure can be arbitrarily complex. Having established that polls are indeed an interesting way to gather data, providing adequate privacy for poll data is the next step. A frequent collector of poll data is the US Census Bureau, who used *differential privacy* (Section 5.2) in their 2020 census [1]. Differential privacy is a rigorous statistical notion of privacy where privacy loss is quantified. Based on their experiences with the 2020 census, the US Census Bureau point out that there exist several issues with differential privacy in a real context, such as I) setting the privacy parameter (ε), and II) the lack of tools to verify the correctness of the implementation of differential privacy [2].

Now, gathering poll data under differential privacy may seem straight forward, but we will argue the devil is in the details. First, there is the detail of the logical representation of a poll. Recalling the poll questions introduced previously, we could add controlled noise to each respondent's answer to achieve differential privacy. Still, if we add this noise naively, we would leak information all the same through the number of responses alone. For example, a person that answers that they were happy with the purchase never gets asked the follow-up question. Hence, the implicit information flow created by follow-up questions is not automatically captured by differential privacy itself. Consequently, we ask ourselves if we can design polls in such a way that the poll's structure does not leak information while still allowing for follow-up questions.

The next detail is in the implementation of the tool. Even when processing data using differentially private query systems, such as PINQ [3] and AIRAVAT [4], information can be leaked through *side-channels* [5]. In these cases, Haeberlen et al. [5] attributed the side-channels to differences in: 1) processing time, and 2) the privacy budget (ε). To complicate our case further, we note that differentially private applications and programming languages (e.g. [6, 7, 3, 8, 9]) tend to focus solely on the *processing* of data, and hence do not address the *collection* of data. As such, an application needs to be put into context so that we can protect against side-channels (Section 5.3).

Lastly, accuracy is an ever important factor in differentially private analyses. As such, we argue that it is important that a tool can provide some measurement of error to address issue I) that was raised by the US Census Bureau. Consequently, we identify three main problems with collecting poll data under local differential privacy:

- Implementation needs to be correct
- Gathered data may be too inaccurate
- Side-channels may arise during collection

To make local differential privacy available for all, we present a novel set of open source tools (Section 5.4) called RANDORI [v] (Japanese:乱取り, meaning: *free-style practice in Japanese martial arts*). RANDORI provides tools for three stages: 1) poll design, 2) tuning the accuracy/privacy trade-off (setting ε), and 3) collecting poll data interactively under local differential privacy. Moreover, we have implemented RANDORI in such a way that we address the logical and implementation details mentioned earlier, which we later evaluate (Section 5.5) and discuss (Section 5.6). We also put our work into context by comparing it to existing research (Section 5.7) before summarizing (Section 5.8).

As such, with RANDORI we focus both on accurate differentially private data

[v] Source code available at:
`https:github.com/niteo/randori`

collection, as well as protecting privacy end-to-end throughout the entire collection process. By presenting RANDORI our contributions are:

+ Tools for *designing polls*, and *collecting data* under differential privacy
+ A tool for *predicting* and *tuning accuracy* for a given poll
+ A data collection process that is *end-to-end private*

5.2 Differential Privacy

Differential privacy is a statistical notion of privacy that represents a property of an algorithm, as opposed to being a property of data. As such, differential privacy is fundamentally different from privacy models such as k-anonymity [10], ℓ-diversity [11] and t-closeness [12], where privacy guarantees are derived from the data.

In this paper we focus specifically on *local differential privacy* [13], which is relevant whenever the mechanism is applied locally at the data source (e.g. a client) rather than centrally (e.g. a server). In the rest of the paper, when we talk about differential privacy, we mean specifically *local* differential privacy.

Definition 1 (ε-Differential Privacy)**.** *A randomized algorithm f, with an input domain $\mathcal{A}$ and an output domain $\mathcal{X}$, is ε-differentially private if for all possible inputs $a, a' \in \mathcal{A}$, and all possible output values $x \in \mathcal{X}$,*

$$Pr[f(a) = x] \leq e^{\varepsilon} \times Pr[f(a') = x].$$

The core mechanism used in this paper to achieve differential privacy is a variant of the classic *randomized response* algorithm [14]. Using a binary input domain ('yes' or 'no'), the randomized response algorithm can be described as follows: flip a coin t. If t lands heads up then respond with the true answer (the input). Otherwise flip a second coin r and return 'yes' if heads, and 'no' if tails. Basically, this algorithm will either deliver the true answer, or randomly choose

one of the viable answers. By delivering an answer in this way, we say that the respondent enjoys *plausible deniability*.

In randomized response the bias of the coin t determines the privacy-accuracy trade-off, whereas the coin r can always be unbiased (i.e. it has a uniform distribution). The variant of this mechanism used in this paper is a simple generalization: it (i) allows for a non-binary input domain, and (ii) permits us to give different accuracy to different answer alternatives.

Definition 2 (Randomized Response)**.** *Let $\mathcal{A}$ be the data domain, and $\mathcal{T} = \{t_a\}_{a \in \mathcal{A}}$ be an indexed set of values in $[0, 1]$. Given these, we define the* randomized response mechanism *$RR_{\mathcal{T}}$, a randomized function from $\mathcal{A}$ to $\mathcal{A}$, as follows:*

$$Pr[RR_{\mathcal{T}}(a) = x] = \begin{cases} t_a + r_a & \textit{when } a = x, \\ r_a & \textit{otherwise.} \end{cases}$$

where $r_a = \frac{1-t_a}{|\mathcal{A}|}$, hence $t_a + r_a \times |\mathcal{A}| = t_a + (1 - t_a) = 1$, and t_a is chosen s.t. $t_a + r_a = Pr[truth] + ((1 - Pr[truth]) \times w_a)$ where w_a is the weights in the tree path to the node containing a.

Now, we also want to be able to reason about the accuracy of our algorithm. Deriving from the concept of a misclassification rate [13], we define our metric for error in Definition 3. That is, if a response a gets mapped by the randomized response mechanism to a value other than itself, it is considered misclassified.

Definition 3 (Error Metric)**.** *Let $RR_{\mathcal{T}}$ represent randomized response, then given for any answer $a \in \mathcal{A}$ the error is the probability of outputting any other output in $\mathcal{A}$:*

$$error_a = Pr[RR_{\mathcal{T}}(a) \neq a]$$

Now, to reason about the accuracy of more than one response we need a notion

that allows us to sum errors. A general analytical error bound [15], for any algorithm, is given by the additive Chernoff bound in Definition 4. We say that an algorithm is (α, β)-useful [16].

Definition 4 (Analytical Accuracy)**.** *Let E be a random variable representing the error of the output of a differentially private algorithm, n is the population size and $\alpha, \beta \in (0, \frac{1}{2})$, where $\beta = 2e^{-2\alpha^2 n}$. Given these, with probability 1-β, the error E is bounded by at most error α:*

$$Pr[E \leq \alpha] \geq 1 - \beta$$

5.3 Threat Model and System Limitations

In order to show that it is possible to build a tool for differentially private data collection that is end-to-end private in a server-client setting, we construct a proof of concept called RANDORI. Our goal is to build a prototype that is I) differentially private by design, II) able to predict error and III) protected against side-channels. In order to make a thorough investigation of side-channels, we introduce a threat model next.

Adversary Model and Assumptions. We assume that adversaries can be either passive or active. The active adversary can send out polls using RANDORI. Consequently, we assume that the adversary can pose as a data analyst. The passive adversary can observe and read the contents of all network traffic between data analyst and respondent. That is, we consider both the case where the communication takes place in plain text, and the case where the adversary is strong enough to break any encryption used during communication. That is, we assume an adversary that can read message contents even when the communication is done over `HTTPS`. Still, we assume that the internal state of the code the respondent is running and the respondent's hardware cannot be monitored by the adversary. That is, the respondent is entering their true answers into a trusted computing base.

We also assume that the respondent does not close their client before our code has finished executing. Later, we elaborate on ways to handle non-termination and the challenges of implementing these defenses in our discussion (Section 5.6).

We do not consider cases where the respondent is an adversary that tries to attack the accuracy of the poll by skewing their answers. That is, we will only consider attacks on privacy, and not attacks on accuracy.

Trust. The sensitive data in this setting is the respondents' *true answers* to polls. That is, *responses* produced by randomized response are not considered sensitive as the respondent enjoys *plausible deniability*. Hence, sensitive data only resides in the respondent's application.

Moreover, we consider the code running on the respondent's device to be completely trusted by the respondent. That is, the code the respondent is running is allowed to hold and take decisions based on sensitive data.

As for the data analysts, we do not consider any of their data to be sensitive. Consequently, the poll questions are considered public data. Hence, the ε for any poll is also public data. We will also assume that the value of each respondent's privacy budget is public. That is, whether or not a respondent has participated in a poll also becomes public. We do not attempt to hide the identity of the respondents, but settle for plausible deniability.

Furthermore, the data analysts are considered untrusted by the respondent. That is, the respondent only wants to share their poll answers under differential privacy, and do not wish to share any other data than what is publicly known with the data analysts.

System Limitations. In RANDORI, the respondents do not have a persistent application. Hence, we cannot store the privacy budget between sessions. Instead, we assume a maximum budget *per poll*. The lack of a persistent application also does not allow us to re-use previous responses to polls as in RAPPOR [17]. This also means that while we are not subject to tracking in the same way RAP-

POR's users are (through re-use of responses), we do not protect against an attacker that sends the same poll twice. As such, we do not address longitudinal privacy in our current proof of concept.

In its current state, RANDORI does not contain a trusted third party to send the RESPONDENT UI to the respondents. Still, adding a third party only requires a minor change where the respondent visits the trusted third party to receive the RESPONDENT UI, but polls can still be served by the untrusted data analyst.

We do not consider the identity of the respondents to be secret, and thus we do not protect against leaking information through participation alone. Also, we do not guarantee security through encryption, since we assume an adversary strong enough to break encryption. Still, we expect the users to encrypt the communication and take adequate measures to store the collected data securely, but we leave this outside of our system scope.

Furthermore, we do not guarantee that RANDORI is optimal from an accuracy or performance aspect.

5.4 Randori

RANDORI is a set of tools with two focal points as far as functionality goes. These focal points are: *poll design* and *data collection*. In this section we will both describe the functionality of RANDORI, the tools it consists of, as well as how differential privacy is achieved. Lastly, we describe the steps taken to assure end-to-end privacy, as this property is not captured by differential privacy itself.

5.4.1 Tools and Vision

RANDORI is a set of tools (Figure 5.1) for designing polls, and for collecting data under differential privacy.

Poll Design: POLL EDITOR

The POLL EDITOR is where the poll structure and content is created and edited. To be able to hide details about the underlying implementation under the hood it consists of two modes: `edit` and `explore`.

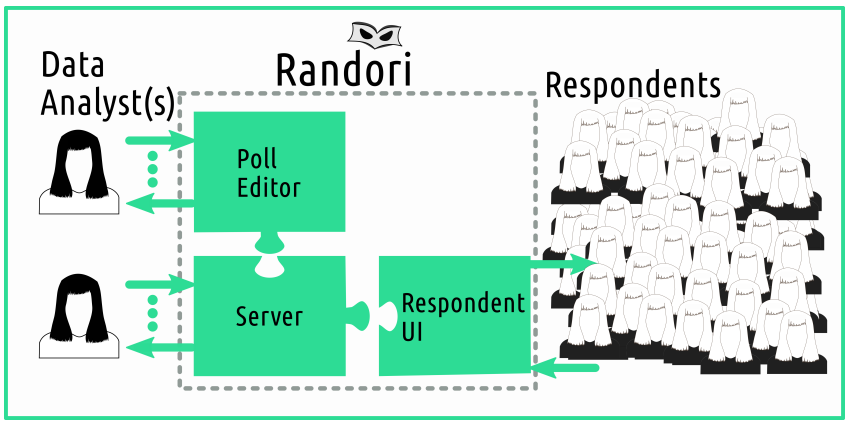


Figure 5.1: The different tools included in RANDORI

In the `edit mode` (screenshot in Appendix Figure 5.7a) the focus is solely on the poll content: order of questions, number of answers and what answers trigger follow-up questions. That is, in the `edit mode` differential privacy is abstracted away. Polls are imported/exported on our `JSON` format (Appendix Listing 5.5).

Then, the fact that data is to be gathered under differential privacy is visible in the `explore mode` (screenshot in Appendix Figure 5.7b). Arguably, without adequate accuracy, collected data becomes useless to the data analyst. To mitigate the problem of high error, we allow for exploration of the accuracy/privacy trade-off through two sets of parameters: (i) `True/Random answers` and `Weight` (corresponding to Pr[truth] and w to calculate t from Definition 2), and (ii) `Alpha`, `Beta` and `Population` (α, β, and n from Definition 4).

The parameters from set (i) influence the value of ε, where the `True/Random`

`answers` slider is a course-grained adjustment affecting *all* answers ($t_1, ..., t_n$ from Definition 2), and weight is a fine-grained adjustment available per answer (affecting a specific t_a). Then, ε is calculated from the poll structure.

All parameters from set (ii) are part of the Chernoff bound and are calculated using Equation (5.1). The parameters `population`, `alpha` and `beta` are shown on a per answer basis. The data analyst is required to set values for two of the parameters per answer, and the POLL EDITOR calculates the third parameter. For example, one can set accuracy (`alpha`) to a fixed value, and then explore which values of `population` and `beta` gives the desired accuracy.

Based on Vadhan [18] we construct the following equation system to display the relationship between α, β, n and ε.

$$\begin{cases} \alpha = \frac{1+e^{\epsilon}}{e^{\epsilon}-1}\lambda \\ \beta = 2e^{-2\lambda^2 n} \\ \varepsilon = \log(\frac{\frac{-\alpha}{\lambda}-1}{1-\frac{\alpha}{\lambda}}) \\ n = \frac{(1+e^{\varepsilon})^2 \log(2/\beta)}{2\alpha^2(e^{\varepsilon}-1)^2} \end{cases} \quad \text{where } \lambda = \sqrt{\frac{\log \frac{2}{\beta}}{2n}} \tag{5.1}$$

Data Collection: SERVER

The SERVER holds the currently active poll on our `JSON` format. The RESPONDENT UI then accesses the poll from e.g. `localhost:5000/poll`. Next, data analysts can access poll results through e.g `localhost:5000/results`. The server post-processes responses from the respondents by filtering away statistical noise using Bayes' theorem. The results are shown to the data analysts in form of a `JSON` file.

Data Collection: RESPONDENT UI

The RESPONDENT UI is a JavaScript client running on the respondents' device. As the RESPONDENT UI is trusted by the respondent, it can branch on

sensitive data to present the respondent with questions based on their previous answers. Hence, to the respondent, a RANDORI polls looks just like any other poll, since randomized response is abstracted away. The RESPONDENT UI also re-calculates ε. Note ε is calculated before the respondent answers the poll and consequently ε does not rely on the respondent's answer.

5.4.2 Differential Privacy

Differential privacy is achieved in RANDORI through randomized response. Since we are in the local setting, ensuring differential privacy is entirely done by the RESPONDENT UI. In particular, we ensure differential privacy through the two following practices:

- Calculation of ε (Section 5.4.2)
- Data representation that prevents information leakage from follow-up questions (Section 5.4.2)

Implementation of Randomized Response

In our implementation of randomized response, the mechanism f_{RR} can be viewed as a stochastic matrix from some input $in \in \mathcal{A}$ to some output $out \in \mathcal{A}$. From Definition 2 and given a biased coin $t_j \in \mathcal{T}$ for each input-output pair, we construct the following stochastic matrix M:

$$M = \begin{array}{c} \text{in} \setminus \text{out} \\ a_1 \\ a_2 \\ \vdots \\ a_{|\mathcal{A}|-1} \\ a_{|\mathcal{A}|} \end{array} \begin{array}{c} \begin{array}{ccccc} a_1 & a_2 & \dots & a_{|\mathcal{A}|-1} & a_{|\mathcal{A}|} \end{array} \\ \begin{pmatrix} t_1 + r_1 & r_1 & \dots & \dots & r_1 \\ r_2 & \ddots & \dots & \dots & \vdots \\ \vdots & \dots & \ddots & \dots & \vdots \\ \vdots & \dots & \dots & \ddots & r_{|\mathcal{A}|-1} \\ r_{|\mathcal{A}|} & \dots & \dots & r_{|\mathcal{A}|} & t_{|\mathcal{A}|} + r_{|\mathcal{A}|} \end{pmatrix} \end{array}$$

Note that each row sums to one by definition of r (Definition 2). Also note that the stochastic matrix is created dynamically based on the size of $\mathcal{A}$ and bias

of the coin. That is, the stochastic matrix does not contain any sensitive data since it only relies on the input from the data analyst. Consequently, we always construct the stochastic matrix before the respondent answers the poll.

Then, to calculate the value of ε for a given stochastic matrix M, we use the following equation:

$$\forall a_j \in \mathcal{A}, \varepsilon = ln\Big(\max\Big(\frac{\min(M_{*j})}{\max(M_{*j})}, \frac{\max(M_{*j})}{\min(M_{*j})}\Big)\Big) \tag{5.2}$$

Where M_{*j} represents a full column in M. Moreover, even when we have non-uniform values for t we do not break differential privacy. Instead, we may end up paying a higher price than what we actually 'use'. As such, we do not suffer from the side-channel that the privacy budget depends on the data as Haeberlen et al. [5] identified in other applications.

Still, for our implementation to be differentially private, we rely on the randomness provided by the programming language. Hence, we strive to make a best effort given the programming language used. Consequently, our implementation of differential privacy uses Javascript's Crypto library to ensure cryptographically strong random values [19], which is the strongest implementation of randomness available in JavaScript. Also related to the implementation of correct randomness is the use of floating point numbers. As incorrect rounding of floats may affect randomness [20], we use fractions in our calculations instead of built-in floats.

To enforce the privacy budget we check that the respondent has enough budget left for the full poll when a poll is received. As this is a proof-of-concept tool as opposed to a monolithic system we have not implemented a stateful client that saves changes to the budget. Instead, we expect practitioners to be able to easily incorporate the RESPONDENT UI into their existing system, and then connecting a budget to a user's profile on their web page.

Lastly, we have also introduced a `truth threshold` (here set to a dummy value of 0.99), as even polls with 100% of truthful answers would otherwise be

considered valid polls. The corresponding validity checks are shown in Listing 5.4.

Mitigating Structural Information Leakage

Next, we address implicit information leaks. In particular, we have investigated how we can allow for follow-up questions without letting them leak information about their parent questions. Recalling our example from the introduction with the question *"How do you feel about your purchase?"*, only respondents that answer '*Unhappy*' get a follow-up question. Accordingly, *any answer to the follow-up question* would leak that the first answer was '*Unhappy*'. We want to ensure that RANDORI can handle follow-up questions without leaking answers to other questions.

Hence, we propose the following representation (Figure 5.2) of a question and its follow-up questions. Note that this representation allows us to always send one response regardless of the tree's depth. That is, the dotted nodes represent answers that the respondent can choose, but that are never sent to the server. Basically, we allow for the respondent to traverse the tree and choose an answer to each triggered question, but in reality the client will only send one response per tree (a question and all its follow-ups). Since the client is a trusted component, branching on secrets is ok, so we can do this without leaking information. In this example, the respondents that choose '*Happy*' or '*Neutral*', are never asked why they were unhappy (which they were not), but the server never learns this due to the use of a single response.

5.4.3 End-to-End Privacy

Simply implementing randomized response to deliver responses is not enough to protect respondents' privacy, since the data collection process may leak additional information. In particular, information leakage through side-channels such as differences in timing, is not captured by randomized response. Consequently, to achieve end-to-end privacy, we need to make sure RANDORI protects against side-channels. To achieve a suitable implementation, we have iterated

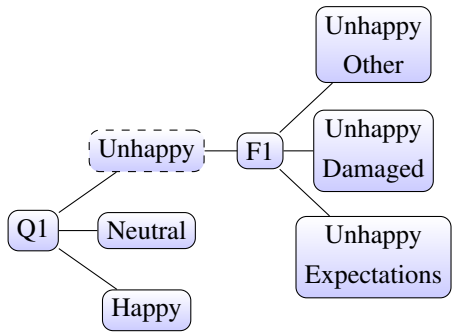


Figure 5.2: Tree structure that allows us to model the behavior of follow-up questions without leaking implicit information

through versions of RESPONDENT UI.

In the end, we arrive at the implementation in Listing 5.1. In order to not let the respondent's answering time leak any information about their answers, we implement a constant time function for answering the poll. This approach of using a constant time function is similar to that of Haeberlen et al. [5]. Essentially, our function depends on `setTimeout`, which sleeps for a given time and then executes the function. By pre-populating *each* question with a random answer, we create a loop that executes in constant time (only depends on the amount of questions, which is not secret).

Unlike Haeberlen et al. [5], we gather data directly from respondents. Consequently, we have an additional attack surface to take into account: namely communication. We will later evaluate which attacks we protect against in Section 5.5.2.

Listing 5.1: Version 4

```
1  var answers = {};
2  var shadow = {};
3  var random = {};
4  var timeout = 9000; //Example
5  fetch('/poll');
6  for(question in poll){
```

```
7       random[question]=random();
8   }
9   setTimeout(submit, timeout);
10  ... // Respondent answers poll
11  function submit(){
12      for(answer in random){
13          if(shadow[answer]==null){
14              answers[answer]=random[answer];
15          } else {
16              answers[answer]=shadow[answer];
17          }
18      }
19      let responses = rr(answers);
20      fetch('/submit', {method: 'POST',
21       body: JSON.stringify(responses)});
22  }
```

5.4.4 Algorithm Summary

In summary, our algorithm performs the following steps sequentially:

1. Flatten poll structure from the `JSON` into one flat tree per question
2. Construct a stochastic matrix M for each tree, then calculate ε using Equation (5.2)
3. Check that respondent has enough budget
4. Check that the `truth threshold` has not been exceeded
5. Deduct ε from the respondent's budget
6. Initiate dummy answers randomly
7. Start timeout
8. On timeout:
    - Lookup the true answer for each question, otherwise use the dummy answer. Next, run randomized response on the answers.

5.5 Privacy Evaluation

In this section, we will show that our implementation is in fact differentially private (Section 5.5.1). Next, we will evaluate end-to-end privacy by investigat-

ing the two attack surfaces available to the adversary: the communication and the content of the response sent (Section 5.5.2).

5.5.1 Differential Privacy

First, we argue that the calculation of ε by the RESPONDENT UI is correct by proving Proposition 1.

Proposition 1. *If M is a stochastic matrix representing some probability function f, then f is ε-differentially private where $e^{\varepsilon} = \max_{i,i',j}\left(\frac{M_{ij}}{M_{i'j}}\right)$.*

Proof. From Definition 1 we require for any two inputs a and a' and any output b

$$\frac{\Pr[f(a)=x]}{\Pr[f(a')=x]} \leq e^{\varepsilon}$$

Let us write M_{ab} for the cell of M representing the probability $\Pr[f(a) = b]$ (recall that a is the row and b is the column in M). By choosing e^{ε} to be the largest value of $\frac{M_{ij}}{M_{i',j}}$ over all choices of i, i', j then clearly

$$\frac{\Pr[f(a)=x]}{\Pr[f(a')=x]} = \frac{M_{ij}}{M_{i',j}} \leq e^{\varepsilon}$$

□

Next, we argue that the inputs are checked and that the privacy budget is correctly enforced. First, the validity of $t_1, ..., t_n$ and $r_1, ..., r_n$ is checked on line 5 and 4 respectively (Listing 5.4). Then, through line 6 (Listing 5.4), respondents' budget threshold is enforced. Since the server is untrusted by the respondent, the client calculates the value of ε from the poll structure (line 3 Listing 5.4). The client will not allow the respondent to answer any question if the respondent cannot afford the full poll (line 7 Listing 5.4). Since we assume the value of a respondent's budget is public information, we do not leak any additional information by not answering due to insufficient budget.

From Section 5.4.2 it is clear that the implicit flow introduced by follow-up questions is mitigated through flattening each question tree. To clarify, since questions with any amount of follow-up questions and questions with no follow-up question both return *exactly one response*, they are indistinguishable to the attacker.

Property	**Implementation**
Validity of poll	Checked on trusted device
Calculation of ε	Follows from Definition 2, and calculated on trusted device
Enforcement of budget	Before poll
Follow-up question triggered or untriggered	Indistinguishable

Table 5.1: Evaluated properties

5.5.2 Side-Channels

Based on our threat model (Section 5.3), the passive adversary can observe and read any network traffic between a data analyst (or an adversary) and a respondent. Since we already explore the implementation of differential privacy, we now assume that all responses sent have *plausible deniability*. Hence, the adversary cannot learn anything about the true answer beyond what ε-differential privacy allows from observing a response.

In order to learn the true answers, the adversary hopes to observe differences in communication or responses and be able to reverse engineer the true answers. Since we are trying to prove the absence of side-channels, our goal is to exhaustively show all possible cases where true answers could cause differences in communication and poll answers, and refute the possibility of them arising in RANDORI. Thus, our objective is to make sure different answers are *indistinguishable* to the adversary.

There are two attack surfaces available to the adversary: the communication itself and the message content. We have identified three cases (Figure 5.3, 5.4 and 5.5), which we walk through next.

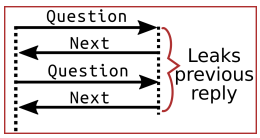


Figure 5.3: **A:** The adversary can learn true answers to questions if a respondent requests (or does not request) follow-up questions

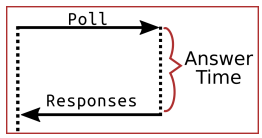


Figure 5.4: **B:** From observing when a poll is received and when the responses are sent, the adversary learns the answering time

```
"Q1" : "Answer"
"F1" : "Answer"
...
"Fn" : "Answer"
"Q2" :
```

Figure 5.5: **C:** Illustration of a poll response with unanswered questions

Case A

There are only two types of communication between the RESPONDENT UI and the SERVER in our implementation: 1) poll `GET` request and 2) response `POST`. We need to make sure that the number of `GET` and `POST` messages are not related to the respondent's true answers.

Mitigation: Always send the full poll. Our implementation does not allow the respondent's client to send any data when requesting a poll (Listing 5.1 line 5) thus requesting anything but the whole poll is impossible. Also, the RESPONDENT UI only replies with one `POST` containing all responses at once. Hence, the scenarios next listed are indistinguishable by design:

- *Respondent requests all questions*
- *Respondent requests some questions*
- *Respondent requests no questions*

Case B

- ▷ **Attack surface:** communication.
- ▷ **Adversary goal:** learn one specific answer.
- ▷ **Example attack:** many follow-ups for *one* specific answer. That is, the adversary will be able to observe differences in how long time it takes for the respondent to finish the poll (Figure 5.4). Here, longer answering time means the follow-up was triggered.

There could be different reasons for differences in answering time, and while it may not be possible for the attacker to say with 100% certainty that the answering time is because of the adversary's follow-ups being triggered, the adversary will be able to shift the probability of their question altering the answer time. Thus, the adversary would be able to gain an advantage.

Consequently, we want to make any reason for differences in timing indistinguishable to an attacker, such that differences in timing do not leak any additional information.

Mitigation: timeout assures constant answering time, since `submit()` is triggered by the timeout (Listing 5.1, line 9). Furthermore, the same amount of instructions are executed (Listing 5.1, line 14 vs line 16) whether the question has been answered or a random pre-populated answer is used. What's more, the for-loop is over `random`, which is of constant size as it contains all question in the poll. Lastly, since the adversary cannot examine the respondent's hardware, they cannot distinguish between the paths in the if-else. Next, we list the differences in timing our implementation takes into account and mitigates:

- ○ *Respondent triggers no follow-ups*
- ○ *Respondent triggers some follow-ups*
- ○ *Respondent triggers all follow-ups*
- ○ *Respondent answers fast, not related to follow-up*
- ○ *Respondent answers slowly, not related to follow-ups*

Case C

- ▷ **Attack surface:** message content.
- ▷ **Adversary goal:** learn one specific answer.
- ▷ **Example attack:** many follow-ups for *one* specific answer which cause the respondent to timeout before answering the last question (Figure 5.5). No answer to the last question means the follow-ups were triggered. Note that this specific attack is introduced by our need to use a timeout.

Since the request for the poll contains no data entered by the respondent, the only place for possible information leakage is through the response `POST`. As each response to a question benefits from plausible deniability due to randomized response, the actual response leaks no information. However, unanswered questions would indeed leak if the respondent answered the question or not. Accordingly, the adversary could learn something by observing how many and which questions are unanswered/answered in the response message.

Mitigation: Since our implementation ensures that each question will have exactly one answer by pre-populating with dummy answers (Listing 5.1 line 12-21), the adversary cannot learn anything new from observing which questions are answered/unanswered. Next, we iterate through all different scenarios where the amount of answered questions could differ:

- ○ *Respondent answers no questions*
- ○ *Respondent answers some questions*
- ○ *Respondent answers all questions*

5.6 Discussion, Limitations and Future Work

Based on the US Census Bureau's experience with differential privacy [1], the main issues with implementing a differentially private poll is ensuring differential privacy and setting ε. We have focused on both these issues by first provid-

ing a proof of concept implementation, and by expressing ε accuracy loss. In our setting, there is also a potential for *dependence between answers* due to the fact that we allow follow-up questions. Consequently, we allow for *non-uniform diagonals* (i.e. different values for t) in our stochastic matrix. While this gives the data analyst more freedom to properly weight their accuracy among answers, it also makes understanding the error more difficult. Hence, we show a Chernoff bound per answer, but this also means that the parameters (α, β, n) also needs to be tweaked per answer. So while we let the data analyst explore the estimated error, we also believe that analytical error bounds may be too blunt for complex polls. Thus, extending RANDORI to include empirical error evaluation remains an open and interesting challenge. In fact, we are currently working on a *simulation environment* that allows for this kind of empirical evaluation.

As for trust, we acknowledge that the respondents receive their client code from the untrusted server. Since the source code of the client is released open source, we assume that the respondent would trust a third party to verify the client code. However, we do not perform any third party checks before letting the respondent answer the poll at this time. A possible and simple extension would be to let a third party serve the client code, and the data analyst would just send the poll.

Regarding the respondent not having a persistent application: this raises two problems. First of all, we do not have a way to save budgets between sessions. We have noted this in our system limitations, but in a real setting this of course becomes a problem, especially when dealing with multiple polls. Our intention is for RANDORI's RESPONDENT UI to be part of an already existing system, for example a web page where the respondent already has an account, which is why we left persistent budgets out of scope. Still, it is important to remember that the respondent's budget needs to be held and updated by a system that the respondent trusts.

Secondly, since the respondent does not have a persistent application, the time-

out fails if the respondent closes their client before timeout happens. When the timeout fails, the analyst will not get any response, and as such the analyst's answer may become less accurate than expected (since the prediction is based on n answers, not $n - 1$). As such the timeout introduces a new problem area: if it is too long, we risk that the participant closes the client too early, and if it is too short, the participant might not have time to answer all questions. We do not provide a definitive answer as to what is the best value for this timeout. The problem is mainly that deciding on an optimal value for a timeout is case dependent, and thus very difficult to give a general answer to. The same problem of setting a timeout arises in Haeberlen et al.'s proposed solution [5]. They [5] argue that if timeouts are chosen properly, decreased accuracy will not happen. Of course, choosing a proper timeout is more tricky in our setting since it involves a real person as opposed to being a case of just approximating a query's execution time.

Another benefit of having a persistent application is that we could re-use responses in the same way RAPPOR [17] does. Still, it would be more tricky to re-use responses in our setting, as it would require us to implement a proper `equals()` function for polls. That is, the question "*How old are you?*" and "*When were you born?*" are semantically similar, but not syntactically similar. Consequently, even if we were to re-use responses to preserve the privacy budget, we may not be able to properly identify which polls should share the same response. As such, re-using responses in our settings requires careful investigation.

Lastly, one entirely unexplored area of RANDORI is usability. So far, we present RANDORI as *a way* to get differential privacy by design, as opposed to *the optimal way*. In our perspective, this paper represents an *in vitro* experiment where we explore how we can provide accuracy and privacy guarantees by design for polls with follow-up question in an interactive setting. Hence, interesting next steps for future work include *user studies* 1) where *real data analysts* collect data using RANDORI, and 2) where we collect data from *real respondents*. In particular, it would be interesting to let the respondents control their own pri-

vacy budget. That is, which values of ε they are comfortable with before they start answering the polls. As of now, the client only calculates the ε, but does not enforce a 'useful' (customized) value of ε in relation to the respondent.

5.7 Related Work

The real world example that is most similar to RANDORI based on what data is collected is the US Census Bureau's deployment of differential privacy [21]. Even though we collect similarly structured data, a big difference is that the Census Bureau's implementation has been tailored to specific data and therefore deploys release mechanisms under centralized differential privacy.

A handful applications have achieved end-to-end privacy by deploying local differential privacy in real settings, for example applications by Google [17], Apple [22, 23, 24] and Microsoft [25]. Out of these, RAPPOR [17] is interesting because they also investigate side-channels. Still, the side-channels identified by Google are different from the ones we face since theirs arise from re-using responses. Another key difference between RANDORI and the aforementioned applications is *how* we choose to gather data. Hence, interacting with respondents and gathering inherently dependent data makes RANDORI novel in comparison.

Also using local differential privacy is the framework PRETPOST [26]. PRETPOST enforces a similar timeout to RANDORI to prevent side-channels, and uses randomized response. So, while the logical implementation of RANDORI and PRETPOST share similarities, RANDORI comes with a graphical interface to tune and evaluate error.

Next up, work that focuses on giving accuracy predictions for differentially private algorithms. First, the Haskell library DPELLA [7] is similar to RANDORI when it comes to our use of Chernoff bounds for exploring accuracy. Still, DPELLA is intended to be used by programmers, and assumes that the data is already stored in a database. DPELLA is also not limited to randomized re-

sponse as opposed to RANDORI.

Lastly, ϵKTELO shares a similar objective with RANDORI as far as providing accurate, differentially private algorithms to users. Noted, ϵKTELO is much more general than RANDORI, and allows users to define new algorithms. What's more, ϵKTELO also concerns the performance of algorithms, which is something we have left completely out of scope in this paper.

5.8 Conclusion

We implement RANDORI, a set of tools for poll design and data collection under differential privacy. A novel part of RANDORI is that we include the data collection process when reasoning about privacy, and hence we also provide defenses against implicit information leakage. What's more, we make RANDORI available for all by releasing it as open source software, in order to motivate uninitiated parties to collect data under differential privacy.

To convince the reader that RANDORI is indeed both differentially private and end-to-end private, we show that our implementation adheres to differential privacy by showing that our algorithm uses a differentially private mechanism. Then, we evaluate and address how we protect polls from implicit information flows. Next, we evaluate end-to-end privacy by systematically walking through each attack surface and eliminate potential attacks. Consequently, through RANDORI, we have made three contributions that map to our originally identified problems. Namely, we provide:

+ tools for *designing polls* and *collecting data* under differential privacy
+ a tool for *predicting and tuning accuracy* of a given poll
+ an *end-to-end private* implementation of randomized response in a server-client setting

Bibliography

[1] M. B. Hawes. “Implementing Differential Privacy: Seven Lessons From the 2020 United States Census”. en. In: *Harvard Data Science Review* 2.2 (Apr. 2020).

[2] S. L. Garfinkel et al. “Issues Encountered Deploying Differential Privacy”. In: *WPES’18*. 2018.

[3] F. 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*. ACM, 2009, pp. 19–30.

[4] I. Roy et al. “Airavat: Security and Privacy for MapReduce”. In: *NSDI*. Proceedings of the 7th Usenix Symposium on Networked Systems Design and Implementation. 2010.

[5] A. Haeberlen et al. “Differential Privacy Under Fire”. en. In: *USENIX Security Symposium* 33 (2011), p. 15.

[6] M. Gaboardi et al. “PSI (Ψ): A Private Data Sharing Interface”. In: *arXiv:1609.04340 [cs, stat]* (Aug. 2018). arXiv: `1609.04340 [cs, stat]`.

[7] E. Lobo-Vesga et al. “A Programming Framework for Differential Privacy with Accuracy Concentration Bounds”. In: *S&P’20*. 2020.

[8] P. Mohan et al. “GUPT: Privacy Preserving Data Analysis Made Easy”. In: *SIGMOD’12*. 2012.

[9] J. Reed and B. C. Pierce. “Distance Makes the Types Grow Stronger: A Calculus for Differential Privacy”. In: *ICFP’10*. 2010.

[10] P. Samarati and L. Sweeney. *Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression*. Tech. rep. SRI International, 1998.

[11] A. Machanavajjhala et al. “L-Diversity: Privacy beyond k -Anonymity”. English. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1.1 (2007).

[12] N. Li et al. “T-Closeness: Privacy Beyond k-Anonymity and l-Diversity”. In: *ICDE ’14*. 2007.

[13] S. Kasiviswanathan et al. “What Can We Learn Privately?” In: *SIAM Journal on Computing* 40.3 (Jan. 2011), pp. 793–826.

[14] S. L. Warner. “Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias”. In: *Journal of the American Statistical Association* 60.309 (Mar. 1965), pp. 63–69.

[15] C. Dwork. “Differential Privacy: A Survey of Results”. In: *Theory and Applications of Models of Computation*. Ed. by M. Agrawal et al. Lecture Notes in Computer Science 4978. Springer Berlin Heidelberg, Jan. 1, 2008, pp. 1–19.

[16] T. Zhu et al. *Differential Privacy and Applications*. Vol. 69. Advances in Information Security. Cham: Springer International Publishing, 2017.

[17] Ú. Erlingsson et al. “RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response”. In: *CCS’14*. 2014.

[18] S. Vadhan. “The Complexity of Differential Privacy”. en. In: *Tutorials on the Foundations of Cryptography*. Ed. by Y. Lindell. Cham: Springer International Publishing, 2017, pp. 347–450.

[19] Mozilla and individual contributors. *Crypto.getRandomValues()*. en. https://developer.mozilla.org/en-US/docs/Web/API/Crypto/getRandomValues. Aug. 2020.

[20] I. Mironov. “On Significance of the Least Significant Bits for Differential Privacy”. In: *Proceedings of the 2012 ACM Conference on Computer and Communications Security*. ACM, Oct. 2012, pp. 650–661.

[21] S. L. Garfinkel. *Report on the 2020 Disclosure Avoidance System as Implemented for the 2018 End-to-End Test*. EN-US. Tech. rep. U.S. Census Bureau, Sept. 2017. Chap. Government.

[22] A. G. Thakurta et al. *Learning New Words*. US Patent US9645998 B1. 2017.

[23] A. G. Thakurta et al. *Emoji Frequency Detection and Deep Link Frequency*. US Patent US9705908 B1. 2017.

[24] Differential Privacy Team, Apple. *Learning with Privacy at Scale*. https://docs-assets.developer.apple.com/ml-research/papers/learning-with-privacy-at-scale.pdf. 2017.

[25] B. Ding et al. "Collecting Telemetry Data Privately". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 3571–3580.

[26] H. Ebadi and D. Sands. *PreTPost: A Transparent, User Verifiable, Local Differential Privacy Framework*. Available online: https://raw.githubusercontent.com/ebadi/preTpost/master/preTpost1.pdf. 2018.

5. Appendix

Listing 5.2: Randomized response as pseudo-code

```
var transitions = pollToMatrix();
function rr(answers){
 for(answer in answers){
  // Find output space
  let outputs = {};
  //Use transitions to get
  // probability per output
  //ranges[include, exclude]->output
  let ranges = {};
  //Use cryptographic random [1,gcd]
  let random = getRandomInt(1,gcd);
  outputs[answer] = ranges[random];
 }
}
```

Listing 5.3: Calculation of ε as pseudo-code

```
var epsilon = undefined;
// For each question subtree
  let potential_epsilon = undefined;
  // For each answer
  let max = undefined;
  let min = undefined;
    // Loop through all other answers
    // Get max probability ratio
    let check=Math.max(max.div(min),
        min.div(max));
    // Bigger?
    if(potential_epsilon==undefined
      || potential_epsilon<check){
      potential_epsilon = check;
    }
  epsilon+=potential_epsilon;
```

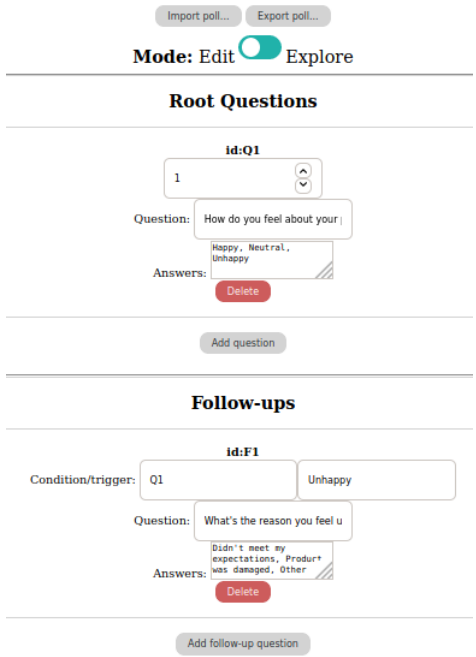
Listing 5.4: Enforcement of budget thershold

```
var budget = 100; // ln(budget)
var truth_threshold = 0.99;
var cost = calculateEpsilon();
var ok_truth = withinThreshold();
if(cost > budget){
  //Disable UI
} else if(!ok_truth){
  //Disable UI
}else {
  budget-=cost;
  // Show poll
}
```

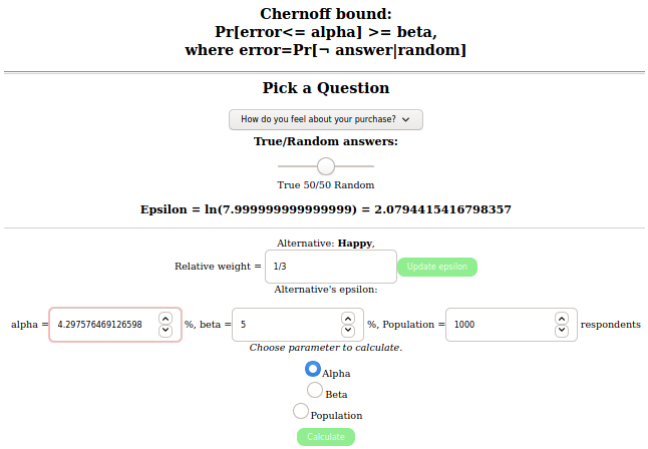
Listing 5.5: JSON format of the example question

```
{"children":[
  {"qid":"F1",
  "question":"What's the reason you feel unhappy?",
  "answers":["Didn't meet my expectations"," Product was damaged"
      ," Other"],
  "probability":["1/3","1/3","1/3"]}
],
"roots":[
  {"qid":"Q1",
  "truth":"1/2",
  "question":"How do you feel about your purchase?",
  "answers":["Happy","Neutral","Unhappy"],
  "probability":["1/3","1/3","1/3"]}
],
"paths":[["Q1","Unhappy","F1"]],
"order":["Q1"]
}
```

Figure 5.6: The JSON produced by the example question



(a) The edit mode of the POLL EDITOR



(b) The explore mode of the POLL EDITOR

Figure 5.7: Graphical interfaces

Paper V

Boel Nelson

Efficient Error Prediction for Differentially Private Algorithms

To appear in *The 16th International Conference on Availability, Reliability and Security (ARES 2021)*, August 17–20, 2021, Vienna, Austria. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3465481.3465746.

Efficient Error Prediction for Differentially Private Algorithms

Abstract

Differential privacy is a strong mathematical notion of privacy. Still, a prominent challenge when using differential privacy in real data collection is understanding and counteracting the accuracy loss that differential privacy imposes. As such, the trade-off of differential privacy needs to be balanced on a case-by-case basis. Applications in the literature tend to focus solely on analytical accuracy bounds, not include data in error prediction, or use arbitrary settings to measure error empirically.

To fill the gap in the literature, we propose a novel application of *factor experiments* to create data aware error predictions. Basically, factor experiments provide a systematic approach to conducting empirical experiments. To demonstrate our methodology in action, we conduct a case study where error is dependent on arbitrarily complex tree structures. We first construct a tool to simulate poll data. Next, we use our simulated data to construct a least squares model to predict error. Last, we show how to validate the model. Consequently, our contribution is a method for constructing error prediction models that are data aware.

6.1 Introduction

Adopting differential privacy in real systems is ultimately an issue of properly understanding the impact differential privacy will have on accuracy. In other words, if an analyst cannot predict the accuracy loss an algorithm will cause, they will be hesitant to use the algorithm. As such, understanding the accuracy loss induced by differential privacy is crucial to differential privacy being deployed in real systems.

In the literature, the accuracy of a differentially private algorithm is often evaluated analytically through Chernoff bounds, such as by Kasiviswanathan et al. [1]. Here, the authors introduce a metric for error, namely misclassification error, which is applicable in their domain. However, the general Chernoff bound they provide requires that there exists a definition for error, i.e. a unit of measurement for the inaccuracy introduced by differential privacy. As such, if the relationship between input variables and error is unknown, Chernoff bounds will not be applicable. As noted by Hay et al. [2], the more complex algorithm, the more difficult it is to analyze the algorithm theoretically. Consequently, some algorithms may be easier to investigate empirically instead of analytically.

In addition, previous research [2, 3] shows that the accuracy of a differentially private algorithm may be greatly influenced by the input data. Consequently, input data should also be taken into account when modeling error. So far, the current literature seems to model error from the algorithm without taking the input data into consideration. For example, Kasiviswanathan et al. [1] and Vadhan [4] use Chernoff bounds, but they do not include input data in their error model.

From the other end of the perspective, several papers including [5, 6, 7, 8, 9, 10, 11] investigate error empirically. Still, input values to the experiments are chosen seemingly arbitrarily. For example, Gao and Ma [10] use {0.005, 0.008,

0.012, 0.015, 0.02} as input values for a threshold variable, and {20, 40, 60, 80, 100} as input for query range size. While these values may be representative for their given domain, this approach requires the authors to rationalize both the chosen ranges and the amount of values used. Furthermore, if a variable is varied in isolation, it is not possible to capture interactions between variables. For example, in [5], the authors vary the number of dimensions, while setting cardinality and ε to fixed values. As such the trend for error when varying the number of dimensions is just captured at a fixed setting.

Hence, we identify three existing problems: 1) the relationship between error and an algorithm's input may be unknown, 2) data oblivious error may result in incorrect error predictions, and 3) choosing representative values for empirical experiments is difficult. To mitigate these problems we propose a novel application of *factor experiments* [12, 13, 14], a statistical approach, to the domain of differential privacy. Here, we show how empirical error measurements can be used to construct an error prediction model using (multiple) linear regression. As such, we are able to model the relationship between all input variables, including data, and error. Accordingly, for the example with ε and population as variables, the prediction model would be in the following format:

$$\begin{aligned} y =& \gamma_0 + \gamma_{threshold} \times \mathsf{threshold} + \gamma_{range} \times \mathsf{range} \\ &+ \gamma_{threshold:range} \times \mathsf{threshold} : \mathsf{range} \end{aligned} \tag{6.3}$$

where y is the predicted error for a specific setting, γ_0 is the intercept, threshold and range are *coded value* representations of the factors, and threshold:range is the possible interaction between factors. Hence, the prediction model is able to predict the error for any value (within the model's span) of threshold and range.

More importantly, factor experiments provide a systematic way to choose the experiment settings where the most information can be extracted. Consequently, our methodology tackles all of the three identified problems by 1) modeling the relationship between variables and error, 2) involving all input variables in model creation, and 3) minimizing the samples required, allowing for efficient

experiments.

We expect our methodology to be valid for any differentially private algorithm: factor experiments allow both numerical and categorical variables, and the analyst may choose any suitable error metric for their domain. To put our methodology into context, we will conduct a case study. In our case study, we run a poll where the algorithm traverses a tree structure before delivering a differentially private reply. Now, we will argue that our use case is particularly interesting in the context of our methodology. First, we have noticed that it is difficult to model the error correctly due to allowing for arbitrarily complex tree structures, where we identify six variables that need to be varied in experiments. Next, it is also difficult to argue for what constitutes a 'good' experiment setting in this case. As such, we believe the many variables' effect on error in our particular use case is difficult to investigate using methods from the current literature. Accordingly, we use RANDORI [15] as a use case where we create a prediction model for error. RANDORI is a set of tools for gathering poll data under *local differential privacy* [16]. So far, RANDORI can predict error analytically through Chernoff bounds, but this error is not data aware. In this paper, we extend RANDORI by adding a simulation tool where users can generate synthetic poll data and empirically evaluate error.

To summarize, prediction models created using our methodology will be able to answer the following questions:

- What is each variable's impact/effect on error?
- Are there any relationships/interactions between variables?

Hence, our contribution is a method for constructing accuracy/error prediction models.

6.2 Background

In this paper, we join two well-known areas: differential privacy and *statistical design of experiments* (DOE) [17]. To provide the reader the necessary background, we describe the trade-off in differential privacy. As we expect our readers to mainly come from the area of differential privacy, we also introduce terminology used in DOE.

6.2.1 Differential Privacy

Differential privacy [18] is a statistical notion of privacy that quantifies the privacy loss. Since differential privacy is a definition and not an implementation, differential privacy can be achieved in different ways, but must always satisfy Definition 2. To define differential privacy, we must first define neighboring data sets (Definition 1).

Definition 1 (Neighboring Data Sets)**.** *Two data sets, D and D', are neighboring if and only if they differ on at most one element d. That is, D' can be constructed from D by adding or removing one single element d:*

$$D' = D \pm d$$

Definition 2 (ε-Differential Privacy)**.** *A randomized algorithm f is ε-differentially private if for all neighboring data sets D, D' and for all sets of outputs $\mathcal{S}$*

$$Pr[f(D) \in \mathcal{S}] \leq exp(\varepsilon) \times Pr[f(D') \in \mathcal{S}]$$

where the probability is taken over the randomness of the algorithm f.

Although differential privacy gives strong mathematical privacy guarantees, implementations introduce some kind of error, relative to an exact but non-private algorithm, to achieve said privacy. The accuracy of a differentially private algorithm can be investigated through analytical accuracy bounds, such as Chernoff bounds. These analytical accuracy bounds are often expressed in general terms,

i.e. they do not define error for a specific algorithm, such as the Chernoff bound given by Kasiviswanathan et al. [1] in Definition 3.

Definition 3 ((α, β)-usefulness)**.** *Let X be a random variable representing the error of the output of a differentially private algorithm f', n is the population size and $\alpha, \beta \in (0, \frac{1}{2})$, where $\beta = 2e^{-2\alpha^2 n}$. Then with probability 1-β, the error X is bounded by at most error α:*

$$Pr[X \leq \alpha] \geq 1 - \beta$$

We say that f' is (α, β)-useful [19].

Note that this formula in particular does not define how to express error. That is, error must be defined on a per-algorithm basis. For example, Kasiviswanathan et al. [1] use misclassification error as their error metric. Still, the resulting accuracy bounds cover the entire *possible* range of error the algorithm can achieve. That is, such theoretical accuracy bounds focus on the worst case error [2]. In other words, the bounds do not describe how error is distributed within the bound. For example, high errors may have very low probability, but an analyst may still condemn the algorithm because the accuracy bounds are not tight enough. Consequently, predicting error using analytical methods can be overly pessimistic.

Furthermore, it can be difficult to properly model the error in order to construct a Chernoff bound. The data dependence of an algorithm's error is particularly important to capture. As Hay et al. [2] point out, a number of differentially private algorithms are indeed data dependent. Hence, data can have an impact on error, but the current literature offers no guidelines on modeling error correctly.

6.2.2 Designed Experiments

In this paper, we will empirically measure the error of a differentially private algorithm. As a consequence, we need to plan and conduct experiments. More specifically, we will conduct *factor experiments* [12, 13], which is a more efficient way of conducting experiments than changing *one factor at a time* (OFAT) [20]. Here, a factor is the same as a variable, and we will use these terms interchangeably.

With factor experiments, we are able to change several factors simultaneously, allowing us to run fewer experiments in total. Essentially, factor experiments is a way of designing experiments such that we can maximize what is learned given a fixed number of measurements [13]. For example, conducting an experiment with two different factors that each can take on 100 different values would require 10 000 measurements with the OFAT approach. Using these same factors but instead running *two-level* factor experiments, we only need to measure the *response* at each edge of the space. That is, only measurements from the black dots in Figure 6.1 are required for factor experiments, whereas the response from each coordinate in the space is required using OFAT.

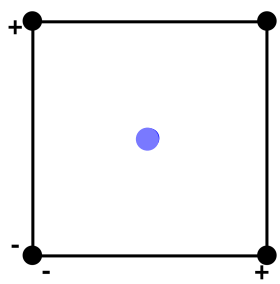


Figure 6.1: The space covered by a factor experiment with two factors. Black dots represents the factors at high/low respectively, and the blue dot is the baseline.

Hence, two-level factor experiments with two factors ($k = 2$) require only $2^k =$

$2^2 = 4$ measurements. In summary, with two-level factor experiments, 2^k measurements are needed for an experiment with k factors. Naturally, factor experiments are much more efficient than OFAT.

When running factor experiments, *coded values* are used to denote *actual values*. For example, two-level experiments usually have a low ('-' or '-1') and a high ('+' or '+1') coded value which are related to the actual values as follows:

$$v_{coded} = \frac{v - a}{b}, \tag{6.4}$$

$$\text{where } a = \frac{v_{high} + v_{low}}{2}, \tag{6.5}$$

$$\text{and } b = \frac{v_{high} - v_{low}}{2} \tag{6.6}$$

So, in a real use case, with the high value (+1) 1000, and the low value (-1) 100, the actual value 500 is represented by the coded value -$\frac{5}{45}$.

Another point of interest in factor experiments is the *baseline*. The baseline is the center point (the blue dot in Figure 6.1) of the entire space that we cover. Consequently, the baseline always has the coded value 0 for each factor.

Using the 2^k responses from the factor experiments, it is possible to construct a prediction model. In this paper, we will construct a linear prediction model using (multiple) linear regression. Given two factors A and B, the linear model can be written as follows:

$$y = \gamma_0 + \gamma_1 A + \gamma_2 B + \gamma_{12} AB + \text{experimental error} \tag{6.7}$$

Where the constant γ_0 is the response at the baseline, and AB is included to capture the possible *interaction* between factor A and B.

Since the prediction model is linear, we will later show how to confirm these assumptions and validate the fit of the model. We also note that in case of non-linear systems, one can instead use three-level factorial designs [21], which are less efficient but are able to capture curvature.

6.3 Methodology

We propose a methodology consisting of four stages:

1. Experiment design
2. Data collection/generation
3. Model creation
4. Model validation

After going through all the stages, the prediction model is ready to be used.

6.3.1 Experiment Design

We propose using two-level factor experiments. This allows linear prediction models to be created. Note that it is important to not choose maximum or minimum values for the levels, as such values likely will be too extreme and not produce a valid model [22]. Instead, choose values that are feasible within the domain. Accordingly, the prediction model will be valid within the space the two levels span, but will not be able to make predictions for values outside. This step is necessary, as extreme values will likely break the assumptions about linearity that allow us to create a linear prediction model.

Next, the k factors involved needs to be identified. This can be done in different ways. The authors note that in software systems, this process is much more straightforward than in for example physical systems, since all possible factors are represented in code. As such, it should be possible to extract all factors from the code directly.

In cases where there are many factors, it might be a good idea to run *screening designs* first, using *fractional designs* [23] experiments to reduce the number of measurements needed. Basically, a fractional design only includes some of the 2^k points, but are chosen in a systematic way. With screening designs, it

is possible to determine if there are factors that can be ignored without running the full 2^k experiments.

Our use case: In RANDORI, data is gathered in poll format. A poll consists of a number of questions and a fixed set of answer alternatives. We represent these questions as trees where a node is either an answer alternative or a question. Furthermore, we also allow follow-up questions in our poll. As such, some answer alternatives have question nodes as children.

Answers to the poll are then gathered using *randomized response* [16]. In randomized response, a respondent will answer truthfully with some probability, Pr[truth], and will otherwise choose a random answer according to a known distribution. In RANDORI, the known distribution is represented through weights attached to each answer alternative.

From our use case, we identify six factors to include in our experiment design. Here, Pr[truth] and relative alternative weight are due to randomized response. Tree depth and Number of alternatives are due to the poll's tree structure. Next, to make our model data aware, we include both the Population and the Number of answers which corresponds to the number of respondents that choose the answer alternative that we target in our measurements. We illustrate all of our identified factors in Figure 6.2. When we measure the error, we will choose one of the alternatives as our target, for example $A1_{Q1}$.

In Table 6.1 we show all our factors and define the levels for each factor.

Now, it makes sense to explain why we have not included ε among our factors. In our case, one thing we want to investigate is the impact of the poll structure on the error. However, there is not a one-to-one mapping between ε and poll structure. That is, while ε can be calculated from the structure of the poll, different structures can result in the same value of ε. As such, only varying ε would not allow us to deduce a unique poll structure.

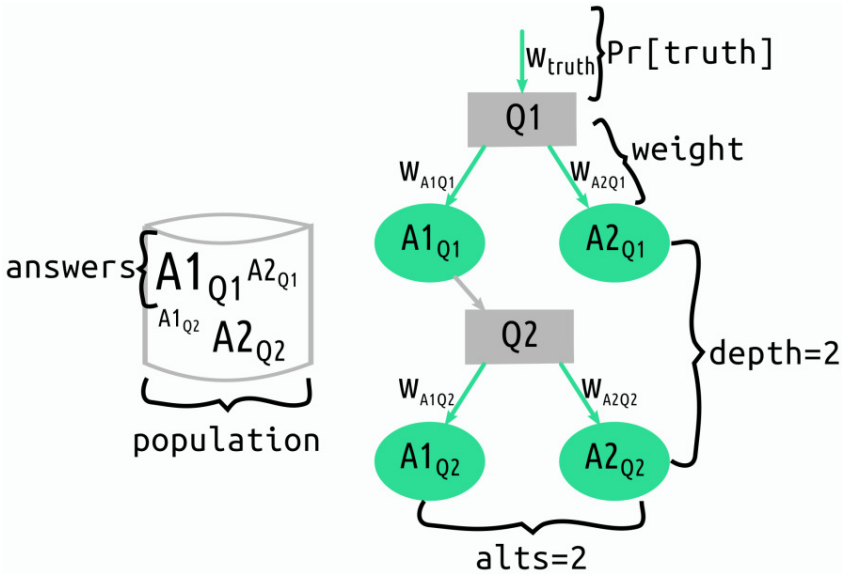


Figure 6.2: The factors used as input to RANDORI, including both data (to the left) and algorithm parameters (to the right). Here, question nodes are gray and answer alternatives are green.

6.3.2 Data Collection/Generation

Data can either be collected from real experiments or generated synthetically. That is, responses from any differentially private algorithm can be used. Note that synthetic data does not make the model less valid: the prediction model will be valid for the entire space covered by the factors. In fact, if the algorithm can be simulated we recommend doing so, as this also eliminates the need to gather potentially sensitive data. Basically, the finesse of factor experiments is that we do not look to sample specific representative settings, but rather we want to be able to cover all values within a known space.

Since results from differentially private algorithms are probabilistic, it is also important to decide whether to measure an average error, or just one measure-

Factor	+	-
Pr[truth]	High	Low
Tree depth	Deep	Shallow
Number of alternatives	Many	Few
Relative alternative weight	High	Low
Population	Many	Few
Number of answers	Many	Few

Table 6.1: Factors, and their respective levels

ment per experiment setting. In this step, it is also important to decide which metric to use for error comparison.

Next, create a table for all the possible combinations of the k factors for a total of 2^k combinations. In physical systems, it is customary to produce the measurements in random order to avoid systematic errors.

Our use case: We construct a tool where we can generate synthetic data and measure the empirical error introduced by randomized response. This tool simulates respondents answering a given poll on RANDORI's format. We call this tool the SIMULATION ENVIRONMENT.

We decide to run each setting 30 times, i.e. $n = 30$, to measure the average error. We also decide to use *mean average percentage error* (MAPE) as our error metric:

$$\text{MAPE} = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{x_t - x'_t}{x_t} \right| \times 100 \tag{6.8}$$

Here, we will calculate the MAPE for one target answer alternative. As such, we measure the distance between the actual percentage (x) of respondents that chose the target alternative, and the estimated percentage (x') calculated from the randomized responses.

6.3.3 Model Creation

From the measured error, it is now possible to create the prediction model. The prediction model is calculated using (multiple) linear regression. To create the prediction model, we suggest using the programming language R. In R, pass the data to the `lm` function and R will output a model. This model will include the effect of each variable and all present interactions between variables.

6.3.4 Model Validation

To test the fit of the model, we first check that the assumptions about linearity hold. Next, the predictions made by the model also need to be investigated. That is, more measurements need to be gathered and compared to the model's predictions for the same setting.

If the model has a good fit, the *residuals* will be small. We use the following formula to calculate the residual r_i when comparing a prediction y_i to a sample measurement s_i for some coordinate i:

$$r_i = y_i - s_i \tag{6.9}$$

A numerical measurement of the model's fit is the (multiple) R^2, the coefficient of determination. A high value of R^2 is necessary but not sufficient for concluding that the fit is good [24]. Next, compare the R^2 value to the adjusted R^2 (calculated as follows: $R^2_{adj} = 1 - (1 - R^2)\frac{N-1}{N-p-1}$, where N is the sample size and p is the number of predictors). The value of R^2 and the adjusted R^2 should be close. Otherwise, a difference indicates that there are terms in the model that are not significant [25]. Consequently, if R^2 and adjusted R^2 differ much, insignificant terms can be removed from the model. In this step, the programming language R can help with providing suggestions for which effects are significant.

Next, we recommend using visual methods to further validate the model due to

NIST's recommendation [26]. These visual methods allow conclusions to be drawn that cannot be drawn from merely observing R^2.

We suggest the following three visual methods:

1. Histogram
2. Residual vs. fitted plot
3. Q-Q normal plot

First, use a histogram to test the residuals for normality. Here, the residuals are expected to have the shape of a normal distribution, and to be centered around 0.

Next, for the residual vs. fitted plot, values should be randomly scattered around 0 on the y-axis [26]. We also expect the *locally weighted scatterplot smoothing* (LOWESS) [27] curve to be flat, since this shows that a linear model is reasonable.

Last, using the Q-Q normal plot shows if the residuals come from a common distribution as the prediction model. If the data sets come from common distributions, the points should be close to the plotted line.

Strategy if the model does not fit: To get quick feedback about the model's fit, pick the three points in Figure 6.3. Next, calculate the residuals for these points.

In cases where the residuals are high, re-use the samples from Figure 6.3 and add the remaining samples needed to create a new, smaller space. That is, systematically zoom in and target a smaller space to make the predictions on. We illustrate this new smaller space in 2D to be able to show a geometric explanation in Figure 6.4.

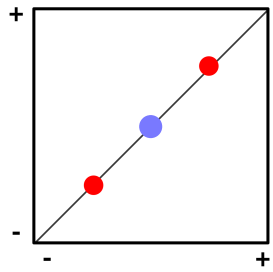


Figure 6.3: The center point, i.e. the baseline represented by the blue dot, and the red dots at (-0.5, -0.5) and (0.5,0.5) respectively

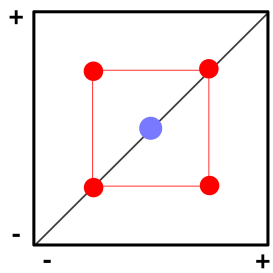


Figure 6.4: Adding the points (0.5,-0.5) and (-0.5,0.5) allows us to zoom in and find a new target space within the red lines

6.4 Results

Next, we will apply our methodology to our use case where we estimate error for poll data. Here, we present the tool we used to generate data (the SIMULATION ENVIRONMENT) and then we show how we iteratively apply the methodology to reach an adequate prediction model.

6.4.1 Simulation Environment

We have built a simulation environment using a Jupyter notebook [28] that takes input on a portable JSON format. The SIMULATION ENVIRONMENT is an

additional tool to the RANDORI [vi] (Figure 6.5) set of open source tools.

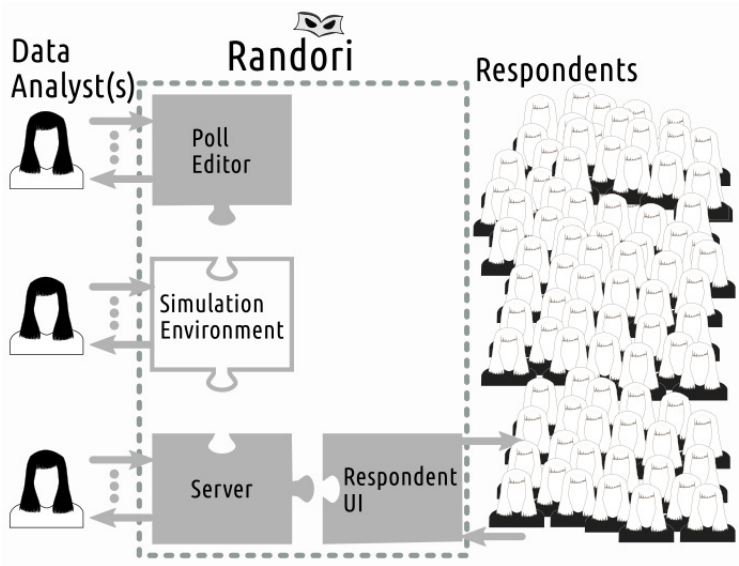


Figure 6.5: The SIMULATION ENVIRONMENT (white puzzle piece) in relation to existing RANDORI tools

Here, a user can construct an arbitrarily complex poll using RANDORI's POLL EDITOR. Next, the poll can be imported into the SIMULATION ENVIRONMENT where the user can tweak all the input variables. In addition, the SIMULATION ENVIRONMENT is able to simulate the respondents' answers either based on probability distributions or a deterministic distribution, although we only use deterministic distributions in this paper.

6.4.2 Experiments

We run a factor experiment with $k = 6$, and calculate the error as MAPE. We run each experiment $n = 30$ times.

[vi] https://github.com/niteo/randori

Using the actual values in Table 6.2 we produce the measurements in Table 6.6 (in Appendix due to length).

Factor	Baseline	+1	-1
Pr[truth]	50%	90%	10%
Tree depth	3	5	1
Number of alternatives	6	10	2
Relative alternative weight	50%	90%	10%
Population	50500	100 000	1000
Number of answers	50%	90%	10%

Table 6.2: Factors and the actual values used for corresponding coded values. In the case of weight and pop the percentage is used for the target alternative, and the remainder is uniformly distributed among siblings.

We enter our data in R and create a prediction model using the `lm` function. Calculating the residual for the baseline, we get a significant error of 384.6646. We pick two additional settings and measure them (Table 6.3) to convince ourselves that the model is indeed a bad fit.

Setting	y_i	s_i	r_i
(0, 0, 0, 0, 0, 0)	418.7087	34.04411	384.6646
(0.5, 0.5, 0.5, 0.5, 0.5, 0.5)	124.8765	14.41732	110.4592
(-0.5, -0.5, -0.5, -0.5, -0.5, -0.5)	731.8813	38.23649	693.6448

Table 6.3: Residuals calculated using the prediction model for the first experiment

As a result, we move on to sample the 2^6 points that covers half the previous space i.e using the settings from Table 6.4. The measured MAPE is in Table 6.7 (in Appendix due to length). We then use these measurements to construct a new prediction model over the smaller space.

Factor	Baseline	+0.5	-0.5
Pr[truth]	50%	70%	30%
Tree depth	3	4	2
Number of alternatives	6	8	4
Relative alternative weight	50%	70%	30%
Population	50500	75750	25250
Number of answers	50%	70%	30%

Table 6.4: Factors and the values used for calculating residuals

From entering our measured values into R's `lm` function, we get a model with 64 coefficients. Using the model, we notice that the prediction for the baseline has improved significantly. The updated prediction is 32.89371, which gives us a residual of $34.04411 - 32.89371 = 1.1504$. Hence, we move on to validate our model.

6.5 Analysis

Next, we move on to validate our model according to our methodology. After validating the model, we will interpret the model.

6.5.1 Evaluating the Model

In order to validate the model, we need to investigate the behavior of the residuals. Hence, we need more measurements. We have decided to pick settings to sample from two sets:

1. The corners (2^6 points) of the middle of the model (like in Figure 6.4) and the center point
2. Any coordinate in the space

We randomly pick 20 points (except that we always include the center point in

the first set) from each of the two approaches, giving us a total of 40 samples to calculate residuals from. Be aware that you may also need to adjust values in certain cases. In our case, we need to take into account that some of our factors are discrete. For example depth is a discrete value and our corner values 0.25 and -0.25 would correspond to a depth of 3.5 and 2.5 respectively. Consequently, we chose to fix depth to 3. The points and their corresponding MAPE is shown in Table 6.5.

	truth	depth	alts	weight	pop	answers	MAPE
0	0	0	0	0	0	0	34.04411
1	0.25	0.00	0.25	0.25	0.25	0.25	20.17603
2	-0.25	0.00	0.25	-0.25	0.25	-0.25	48.18286
3	0.25	0.00	-0.25	-0.25	0.25	-0.25	31.06755
4	-0.25	0.00	0.25	-0.25	-0.25	0.25	50.33476
5	0.25	0.00	-0.25	0.25	0.25	0.25	19.59611
6	-0.25	0.00	0.25	0.25	0.25	0.25	27.66037
7	-0.25	0.00	-0.25	-0.25	0.25	-0.25	46.24753
8	-0.25	0.00	-0.25	0.25	0.25	0.25	26.60268
9	0.25	0.00	-0.25	0.25	0.25	-0.25	17.30670
10	-0.25	0.00	0.25	0.25	-0.25	-0.25	25.07704
11	-0.25	0.00	-0.25	-0.25	0.25	-0.25	46.36067
12	-0.25	0.00	-0.25	-0.25	0.25	-0.25	46.18749
13	0.25	0.00	-0.25	0.25	-0.25	0.25	19.71108
14	0.25	0.00	-0.25	-0.25	-0.25	0.25	33.26383
15	-0.25	0.00	0.25	-0.25	-0.25	-0.25	48.09976
16	-0.25	0.00	0.25	0.25	-0.25	0.25	27.58968
17	-0.25	0.00	-0.25	0.25	0.25	-0.25	22.55290
18	-0.25	0.00	0.25	0.25	0.25	-0.25	24.97823
19	0.25	0.00	-0.25	0.25	0.25	0.25	19.61443
20	-0.50	-0.50	-0.50	0.03	-0.46	0.28	8.42964
21	0.16	0.25	0.00	0.32	-0.25	0.38	28.34642
22	-0.06	-0.25	-0.50	0.03	-0.31	-0.32	8.82148

	truth	depth	alts	weight	pop	answers	MAPE
23	-0.50	0.25	-0.25	0.03	0.03	-0.29	53.20864
24	0.21	0.50	0.00	0.12	-0.17	0.34	36.71494
25	0.31	0.50	0.25	0.34	-0.02	0.39	29.04886
26	-0.49	0.25	0.25	-0.22	-0.12	0.07	63.40224
27	-0.27	-0.50	0.00	0.35	0.29	0.34	65.43967
28	0.39	0.25	0.50	0.21	-0.03	0.38	25.73380
29	0.39	-0.25	0.00	0.30	0.13	0.28	3.46581
30	-0.45	0.50	0.50	0.06	-0.04	-0.21	59.91642
31	-0.00	0.50	-0.25	-0.36	0.05	-0.02	47.62934
32	-0.20	-0.25	-0.50	-0.03	0.16	0.42	21.80034
33	-0.14	0.25	0.50	-0.40	0.11	0.46	53.57877
34	0.11	0.00	-0.25	-0.48	-0.35	-0.21	39.38831
35	0.14	0.00	0.00	-0.37	0.15	0.02	38.41253
36	-0.09	-0.50	-0.50	-0.41	-0.47	-0.39	5.75857
37	-0.19	0.50	0.25	-0.08	0.44	-0.19	52.70103
38	0.42	-0.50	-0.25	-0.19	0.00	-0.01	2.18997
39	-0.47	0.50	-0.25	0.33	-0.33	0.35	51.42151

Table 6.5: The sampled points used and their measured MAPE

First, we check the value of our R^2. For our model, the R^2 is 0.8419. However, we notice that the adjusted R^2 is significantly lower, 0.5929. Seeing as we have 64 coefficients, it seems reasonable to simplify our model to avoid overfitting. We update our model in R to only involve the effects that R marks as significant. To do this, we enter the suggested effects in R, which in our case are:

```
lm(formula = MAPE ~ truth + alts + weight +
truth*depth+depth*weight + truth*depth*weight +
depth*weight*answers ).
```

Now, we end up with a R^2 of 0.7846, and an adjusted R^2 of 0.7562. These

values are still high, and since they are now significantly closer, we move on to validate the model visually.

Next, we plot the residuals as a histogram in Figure 6.6. From the histogram, we see that our residuals are indeed centered around 0. The histogram indicates a normal distribution. Hence, we move on to the next test.

Figure 6.6: A histogram of the residuals

Now, we want to investigate the relationship between fitted values (measurements) and the model's prediction. Then, we plot fitted values vs. predictions in Figure 6.7. We observe that the residuals appear to not have a specific shape around the y-axis. We also conclude that the LOWESS fit curve appears to be almost flat.

Finally, we investigate the normal Q-Q plot (Figure 6.8). We see that most points follow the plotted line, indicating that our predictions come from the same distribution as the measured values. Hence, we conclude that our prediction model is valid for our use case.

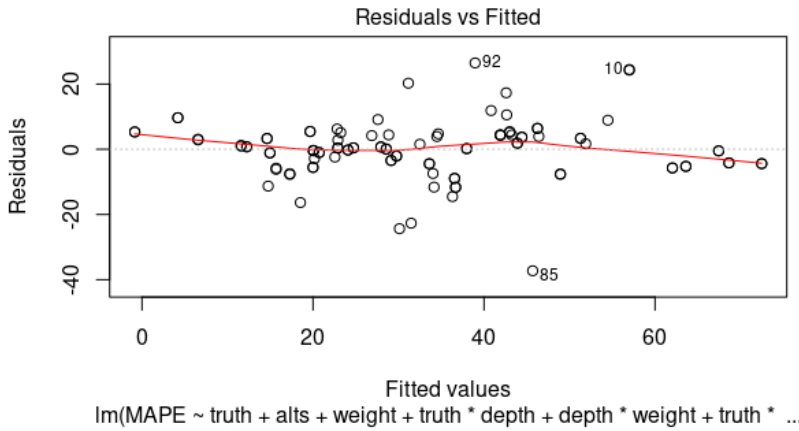


Figure 6.7: Residuals represented as circles, fitted values as the dotted line. The red line represents the LOWESS fit of the residuals vs. fitted values.

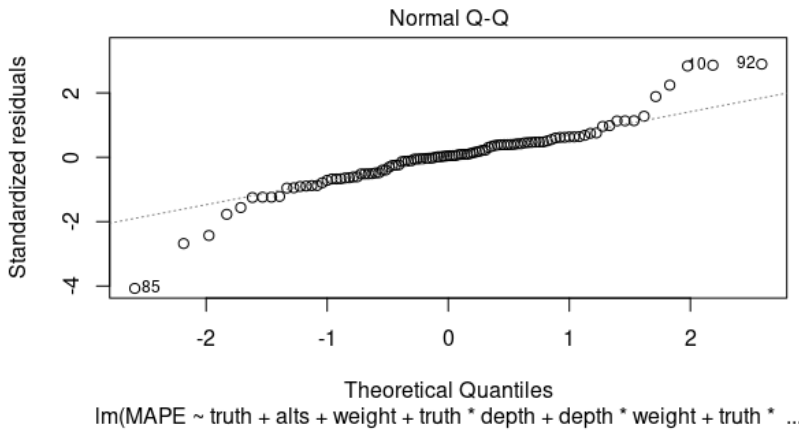


Figure 6.8: The normal quantile-quantile plot

6.5.2 Interpreting the Model

The model is now ready to be used. That is, any value within each factor's range [high,low] can be plugged in to produce an error prediction. It is also possible to set $y \leq c$, with c being our maximum tolerable error, and then find which settings satisfy the inequality. Our final error prediction model is as follows:

$$\begin{aligned} y = {} & 32.501266 - 29.023493 \times \textsf{truth} + 5.037411 \times \textsf{alts} \\ & - 16.562410 \times \textsf{weight} + 1.449934 \times \textsf{depth} \\ & + 1.856916 \times \textsf{answers} + 10.044302 \times \textsf{truth : depth} \\ & - 28.397984 \times \textsf{weight : depth} \\ & + 4.175231 \times \textsf{truth : weight} \\ & + 8.535667 \times \textsf{depth : answers} \\ & - 8.402531 \times \textsf{weight : answers} \\ & + 51.134829 \times \textsf{truth : weight : depth} \\ & + 25.945740 \times \textsf{weight : depth : answers} \end{aligned} \tag{6.10}$$

We note that the simplification step has allowed us to completely eliminate pop from our factors. As such, we draw the conclusion that the population size itself does not have a significant impact on error.

To get an overview of our model, we use a Pareto plot [29] (Figure 6.9) which allows us to visually compare all effects at once. Here, effects are ordered by magnitude.

From the plot, it is clear that truth:weight:depth affects error the most. Maybe most notably, truth:weight:depth increases error whereas its components truth and weight:depth both decrease error. From examining the Pareto plot, it seems that truth:weight is the interaction that causes the increase in error.

As expected, truth has a negative impact on error. That is, a high value of truth

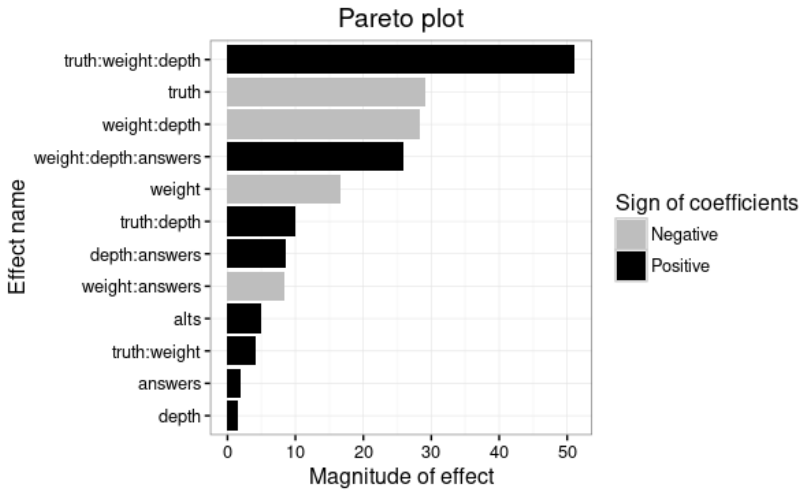


Figure 6.9: The Pareto plot of the simplified model

will reduce error. More surprisingly, truth is involved in several interactions which all increase error.

It may be tempting to completely ignore answers and depth as these two factors have the lowest magnitude of effect. However, ignoring these factors is dangerous: they are both involved in interactions that have significantly higher magnitude.

The factor alts is the only one that does not have interactions. It may seem counter-intuitive that having more siblings have such a small impact on error. Still, the magnitude of this effect may very well be due to our choice to input polls where we uniformly distribute the remaining weight among the siblings.

Hence, we can give RANDORI's users the following advice: use the model to find local minima or ranges. The model can also be used to find minima and ranges while accounting for known constraints such as for example

Pr[truth] ≤ 0.5. When working in RANDORI's POLL EDITOR it is important to beware of the main effect truth:weight:depth and its building blocks. As weight primarily is involved in decreasing error, we recommend increasing weight before tweaking the other factors.

6.6 Discussion, Limitations and Future Work

A limitation in our work is that the prediction models we create are linear. As such, prediction can be off in cases where the error is in fact non-linear. Still, factor experiments can nevertheless be used to make predictions for non-linear systems. To facilitate for non-linear systems the factor levels have to be chosen differently: i.e. we would need 3 levels [21] instead of 2. Hence, our approach can be adapted to create non-linear models by running more experiments.

Additionally, we know that error should also depend on the, non-linear, term $exp(\varepsilon)$ from the definition of differential privacy. Still, it is not clear how the term $exp(\varepsilon)$ and other, algorithm specific, factors compare in order of magnitude. As such, more research is needed to see if ε can be modeled in a suitable way, or if perhaps ε needs to be transformed to be linear ($ln(exp(\varepsilon))$. Nevertheless, factor experiments still provide a systematic and efficient way to explore the impact of different variables on error. That is, factor experiments may still be used to explore the other factors' impact on error. Hence, while it may not always be possible to extract an accurate prediction model, factor experiments are still useful when determining which data points should be used as input to test the accuracy of a differentially private algorithm.

Furthermore, factor experiments provide a possible way to systematically predict error for *all* representative input data sets for a differentially private algorithm. That is, instead of using real data sets to predict error, factor experiments statistically emulate all possible data sets bounded by the experiment's levels (the high/low values for each variable in our case). Hence, using factor experi-

ments to create prediction models can be more robust statistically than making predictions based on one real data set.

Whether the model is correct or not will be identified when testing the model according to our methodology. If the model is incorrect it can be due to error being non-linear, but it can also be due to not including all relevant factors. As such, an incorrect model requires further investigation.

Accordingly, correctly identifying relevant factors is crucial to building a correct model. Still, there exists no recognized way of correctly and efficiently identifying all factors. As mentioned in Section 6.3.1, it is nonetheless possible to try if a factor is relevant using *screening designs* before running a full factorial experiment. From our use case, it is nonetheless clear that some candidate factors rule themselves out by simply being impossible to implement. For example, we considered having the factor *number of parent siblings* together with depth, which results in the impossible combination of having no parents (depth=0) and also having parent siblings. Hence, we believe looking for possible contradictions among factors is important when designing the experiments.

In order to not create contradicting factors, we have also decided to only model the weight for the target alternative. That is, we set the weight for the target alternative (or the target's parent), and uniformly divide the remainder among the siblings. For example, when a target has weight 70% and three siblings, each sibling gets $\frac{100-70}{3}\%$ each. As such, we have not investigated settings where the siblings have non-uniform weight distributions.

One decision that may seem controversial is that we do not include ε as one of the factors in our model. While we do not tweak ε directly, we do in fact adjust ε by changing the structure of the poll. The reason we have chosen to indirectly tweak ε as to tweaking it directly is that one single value of ε corresponds to multiple poll structures, whereas one poll structure corresponds to exactly one value of ε. Hence, while it may seem unintuitive at first, indirectly tweaking ε

makes more sense than tweaking it directly in our case.

Somewhat surprising is that population was eliminated from our prediction model in the simplification step. We argue that the elimination of population is because answers is related to pop (the probability of choosing some alternative Ai_{Qj} is Pr[Ai_{Qj}] = pop*answers), and population therefore becomes redundant. It is also possible that the choice of error measurement, MAPE in our case, contributes to making population irrelevant since it is a relative measurement of error as opposed to an absolute measurement.

Finally, we note that in this paper we have measured the error of leaf nodes in a tree. Still, with the known relationships between answers, it appears to be possible to further post-process and add accuracy to parent answers. We believe including the number of children as a factor would be an interesting path to explore next in order to better understand the effect of this post-processing. Put differently, the challenge here is properly modeling the factors without creating contradictions between factors.

6.7 Related Work

As mentioned in Section 6.1, evaluating error empirically is not a new topic within differential privacy. However, creating prediction models from empirical data appears to be a novel approach.

The work closest to ours is DPBENCH [2], which is an error evaluation framework for differentially private algorithms. In DPBENCH, the authors propose a set of *evaluation principles*, including guidelines for creating diverse input for algorithms. Hence, DPBENCH has a strong focus on understanding the data-dependence of an algorithm's error. Still, DPBENCH does not produce an error prediction model like we do, nor does it minimize the number of experiments needed to conduct.

We also note that DPCOMP [3] is the closest work to our SIMULATION ENVIRONMENT. DPCOMP allows users to compare how the accuracy of a differentially private algorithm is affected by varying input data. Our work is similar in the sense that our SIMULATION ENVIRONMENT also is intended to be used to evaluate trade-offs. Our SIMULATION ENVIRONMENT is also inspired by DPBENCH's evaluation principles and consequently allows data following different distributions to be entered and evaluated. However, our simulation environment is less general than DPCOMP, since our solution uses one fixed algorithm.

6.8 Conclusion

We have presented a methodology for empirically estimating error in differentially private algorithms which 1) models the relationships between input parameters, 2) is data aware, and 3) minimizes the measurements required as input. Hence, prediction models created using our methodology allow for expressive, data aware, error prediction. Moreover, we conducted a case study where we apply our methodology to a setting where error is measured from poll structures. To support our use case, we have added a simulation tool to the RANDORI open source tool suite, adding the functionality of generating synthetic data and evaluating error empirically.

From our case study, we were able to create a prediction model for error using six factors. After evaluating and simplifying our model, we are able to answer the two questions from our introduction. First, there are 13 main effects on error. Next, there are seven interactions.

From evaluating the prediction model we found that our model has a good fit. As such, our novel application of factor experiments shows promising results as a methodology for error evaluation of differentially private algorithms.

Consequently, we have contributed with a novel application of a methodology

that shows promise for error prediction of differentially private algorithms. In addition, we have also built a simulation environment that generates synthetic poll data and measures error through simulating randomized response.

One interesting path for future work is to investigate if, and how, the number of factors used in the model prediction affects the model's fit. Along a similar line of thought, it would also be interesting to attempt to create prediction models for well known differentially private algorithms and libraries. As such, we encourage the use of our methodology in order to construct error prediction models for other differentially private algorithms.

6.8 Acknowledgements

This work was partly funded by the Swedish Foundation for Strategic Research (SSF) and the Swedish Research Council (VR).

Bibliography

[1] S. Kasiviswanathan et al. “What Can We Learn Privately?” In: *SIAM Journal on Computing* 40.3 (Jan. 2011), pp. 793–826.

[2] M. Hay et al. “Principled Evaluation of Differentially Private Algorithms Using DPBench”. In: *Proceedings of the 2016 International Conference on Management of Data*. SIGMOD ’16. New York, NY, USA: ACM, 2016, pp. 139–154.

[3] M. Hay et al. “Exploring Privacy-Accuracy Tradeoffs Using DPComp”. In: *SIGMOD ’16*. 2016.

[4] S. Vadhan. “The Complexity of Differential Privacy”. en. In: *Tutorials on the Foundations of Cryptography*. Ed. by Y. Lindell. Cham: Springer International Publishing, 2017, pp. 347–450.

[5] B. Ding et al. “Differentially Private Data Cubes: Optimizing Noise Sources and Consistency”. In: *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*. SIGMOD ’11. New York, NY, USA: ACM, 2011, pp. 217–228.

[6] Y. Xiao et al. “DPCube: Releasing Differentially Private Data Cubes for Health Information”. In: *International Conference on Data Engineering (ICDE)*. Arlington, VA, USA: IEEE, Apr. 2012, pp. 1305–1308.

[7] H. Li et al. “Differentially Private Histogram Publication for Dynamic Datasets: An Adaptive Sampling Approach”. In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge*

Management. CIKM '15. Melbourne, Australia: Association for Computing Machinery, 2015, pp. 1001–1010. URL: https://doi.org/10.1145/2806416.2806441.

[8] R. Chen et al. "Private Analysis of Infinite Data Streams via Retroactive Grouping". In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. CIKM '15. Melbourne, Australia: Association for Computing Machinery, 2015, pp. 1061–1070. URL: https://doi.org/10.1145/2806416.2806454.

[9] T. Benkhelif et al. "Co-Clustering for Differentially Private Synthetic Data Generation". In: *Personal Analytics and Privacy. An Individual and Collective Perspective*. Ed. by R. Guidotti et al. Cham: Springer International Publishing, 2017, pp. 36–47.

[10] R. Gao and X. Ma. "Dynamic Data Histogram Publishing Based on Differential Privacy". In: *2018 IEEE Intl Conf on Parallel Distributed Processing with Applications, Ubiquitous Computing Communications, Big Data Cloud Computing, Social Computing Networking, Sustainable Computing Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*. Melbourne, VIC, Australia: IEEE, 2018, pp. 737–743.

[11] H. Li et al. "IHP: Improving the Utility in Differential Private Histogram Publication". In: *Distributed and Parallel Databases* 37 (2019), pp. 721–750.

[12] NIST/SEMATECH. *NIST/SEMATECH e-Handbook of Statistical Methods*. https://www.itl.nist.gov/div898/handbook/index.htm. [Accessed: 2021-02-17]. 2013.

[13] NIST/SEMATECH. *5.1.1. What Is Experimental Design?* https://www.itl.nist.gov/div898/handbook//pri/section1/pri11.htm. [Accessed: 2021-02-17]. 2013.

[14] NIST/SEMATECH. *5.3.3.3. Full Factorial Designs*. https://www.itl.nist.gov/div898/handbook/pri/section3/pri333.htm. [Accessed: 2021-02-24]. 2013.

[15] B. Nelson. “Randori: Local Differential Privacy for All”. In: *arXiv:2101.11502 [cs]* (Jan. 2021). arXiv: 2101.11502 [cs].

[16] S. L. Warner. “Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias”. In: *Journal of the American Statistical Association* 60.309 (Mar. 1965), pp. 63–69.

[17] NIST/SEMATECH. *4.3.1. What Is Design of Experiments (DOE)?* https://www.itl.nist.gov/div898/handbook/pmd/section3/pmd31.htm. [Accessed: 2021-03-02]. 2013.

[18] C. Dwork et al. “Calibrating Noise to Sensitivity in Private Data Analysis”. In: *Theory of Cryptography*. Ed. by S. Halevi and T. Rabin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 265–284.

[19] T. Zhu et al. *Differential Privacy and Applications*. Vol. 69. Advances in Information Security. Cham: Springer International Publishing, 2017.

[20] R. A. Fisher. *Statistical Methods, Experimental Design, and Scientific Inference*. English. 1st edition. Oxford, England: Oxford University Press, Aug. 1990.

[21] NIST/SEMATECH. *5.3.3.9. Three-Level Full Factorial Designs*. https://www.itl.nist.gov/div898/handbook//pri/section3/pri339.htm. [Accessed: 2021-05-15]. 2013.

[22] K. Dunn. *Process Improvement Using Data*. en. Release 0f428b. Jan. 2021.

[23] NIST/SEMATECH. *5.3.3.4. Fractional Factorial Designs*. https://www.itl.nist.gov/div898/handbook/pri/section3/pri334.htm. [Accessed: 2021-02-24]. 2013.

[24] NIST/SEMATECH. *4.4.4. How Can I Tell If a Model Fits My Data?* https://www.itl.nist.gov/div898/handbook/pmd/section4/pmd44.htm. [Accessed: 2021-02-24]. 2013.

[25] NIST/SEMATECH. *5.4.4. How to Test and Revise DOE Models*. https://www.itl.nist.gov/div898/handbook/pri/section4/pri44.htm. [Accessed: 2021-02-24]. 2013.

[26] NIST/SEMATECH. *5.2.4. Are the Model Residuals Well-Behaved?* https://www.itl.nist.gov/div898/handbook/pri/section2/pri24.htm. [Accessed: 2021-02-17]. 2013.

[27] NIST/SEMATECH. *4.1.4.4. LOESS (Aka LOWESS)*. https://www.itl.nist.gov/div898/handbook/pmd/section1/pmd144.htm. [Accessed: 2021-05-15]. 2013.

[28] Project Jupyter. *Project Jupyter*. https://www.jupyter.org. [Accessed: 2021-05-15]. 2021.

[29] RDocumentation. *Pareto.Chart Function - RDocumentation*. https://www.rdocumentation.org/packages/qcc/versions/2.6/topics/pareto.chart. [Accessed: 2021-05-15]. 2021.

6.A Experiment Details

Standard order	Pr[truth]	Tree depth	Number of alternatives	Alternative weight	Population	Number of answers	MAPE
N/A	0	0	0	0	0	0	34.04411
1	-	-	-	-	-	-	87.08667
2	+	-	-	-	-	-	3.49111
3	-	+	-	-	-	-	37.57905
4	+	+	-	-	-	-	4.90007
5	-	-	+	-	-	-	47.75
6	+	-	+	-	-	-	6.58
7	-	+	+	-	-	-	76.73124
8	+	+	+	-	-	-	8.56657
9	-	-	-	+	-	-	7365.33667
10	+	-	-	+	-	-	96.20333
11	-	+	-	+	-	-	1228.76234
12	+	+	-	+	-	-	19.77456
13	-	-	+	+	-	-	1456.40333

Standard order	Pr[truth]	Tree depth	Number of alternatives	Alternative weight	Population	Number of answers	MAPE
14	+	-	+	+	-	-	18.47
15	-	+	+	+	-	-	405.1528
16	+	+	+	+	-	-	3.74374
17	-	-	-	-	+	-	90.03673
18	+	-	-	-	+	-	1.21997
19	-	+	-	-	+	-	39.38121
20	+	+	-	-	+	-	4.38645
21	-	-	+	-	+	-	47.13567
22	+	-	+	-	+	-	7.02496
23	-	+	+	-	+	-	75.60747
24	+	+	+	-	+	-	8.34256
25	-	-	-	+	+	-	7362.4095
26	+	-	-	+	+	-	98.25777
27	-	+	-	+	+	-	1240.11986
28	+	+	-	+	+	-	19.7394
29	-	-	+	+	+	-	1466.18583
30	+	-	+	+	+	-	18.8858
31	-	+	+	+	+	-	403.33846
32	+	+	+	+	+	-	4.16551
33	-	-	-	-	-	+	61.83111
34	+	-	-	-	-	+	8.08626
35	-	+	-	-	-	+	88.29154
36	+	+	-	-	-	+	9.66657
37	-	-	+	-	-	+	63.75222
38	+	-	+	-	-	+	8.2323
39	-	+	+	-	-	+	89.69907
40	+	+	+	-	-	+	10.02583
41	-	-	-	+	-	+	811.41556
42	+	-	-	+	-	+	10.13037

Standard order	Pr[truth]	Tree depth	Number of alternatives	Alternative weight	Population	Number of answers	MAPE
43	-	+	-	+	-	+	310.01569
44	+	+	-	+	-	+	2.16437
45	-	-	+	+	-	+	738.71667
46	+	-	+	+	-	+	9.07111
47	-	+	+	+	-	+	300.02957
48	+	+	+	+	-	+	2.1328
49	-	-	-	-	+	+	61.99979
50	+	-	-	-	+	+	7.9004
51	-	+	-	-	+	+	88.42618
52	+	+	-	-	+	+	9.84616
53	-	-	+	-	+	+	63.75659
54	+	-	+	-	+	+	7.95395
55	-	+	+	-	+	+	89.82931
56	+	+	+	-	+	+	9.95786
57	-	-	-	+	+	+	810.22851
58	+	-	-	+	+	+	9.99809
59	-	+	-	+	+	+	310.55943
60	+	+	-	+	+	+	2.44021
61	-	-	+	+	+	+	737.21517
62	-	+	+	+	+	+	299.99379
63	+	-	+	+	+	+	9.01693
64	+	+	+	+	+	+	2.20558

Table 6.6: MAPE measurements for the experiment using -1 and +1 as coded value inputs

Standard order	Pr[truth]	Tree depth	Number of alternatives	Alternative weight	Population	Number of answers	MAPE
N/A	0	0	0	0	0	0	34.04411
1	-	-	-	-	-	-	38.23649
2	+	-	-	-	-	-	17.89185
3	-	+	-	-	-	-	58.33831
4	+	+	-	-	-	-	25.18673
5	-	-	+	-	-	-	48.15875
6	+	-	+	-	-	-	25.15229
7	-	+	+	-	-	-	64.44095
8	+	+	+	-	-	-	27.66351
9	-	-	-	+	-	-	81.467
10	+	-	-	+	-	-	13.00362
11	-	+	-	+	-	-	9.89232
12	+	+	-	+	-	-	9.41709
13	-	-	+	+	-	-	56.28555
14	+	-	+	+	-	-	9.56171
15	-	+	+	+	-	-	19.75423
16	+	+	+	+	-	-	12.79737
17	-	-	-	-	+	-	38.11988
18	+	-	-	-	+	-	17.97198
19	-	+	-	-	+	-	58.37657
20	+	+	-	-	+	-	25.14935
21	-	-	+	-	+	-	48.43102
22	+	-	+	-	+	-	25.08915
23	-	+	+	-	+	-	64.49147
24	+	+	+	-	+	-	27.73975
25	-	-	-	+	+	-	81.24882
26	+	-	-	+	+	-	13.02403
27	-	+	-	+	+	-	9.5234
28	+	+	-	+	+	-	9.65797

Standard order	Pr[truth]	Tree depth	Number of alternatives	Alternative weight	Population	Number of answers	MAPE
29	-	-	+	+	+	-	56.3261
30	+	-	+	+	+	-	9.79661
31	-	+	+	+	+	-	19.70136
32	+	+	+	+	+	-	12.57202
33	-	-	-	-	-	+	52.6255
34	+	-	-	-	-	+	23.3408
35	-	+	-	-	-	+	66.96285
36	+	+	-	-	-	+	28.56059
37	-	-	+	-	-	+	54.63909
38	+	-	+	-	-	+	28.61188
39	-	+	+	-	-	+	68.09695
40	+	+	+	-	-	+	29.17961
41	-	-	-	+	-	+	45.78992
42	+	-	-	+	-	+	4.45637
43	-	+	-	+	-	+	23.87327
44	+	+	-	+	-	+	13.78785
45	-	-	+	+	-	+	41.37552
46	+	-	+	+	-	+	13.85628
47	-	+	+	+	-	+	25.68611
48	+	+	+	+	-	+	14.47902
49	-	-	-	-	+	+	52.71001
50	+	-	-	-	+	+	23.2522
51	-	+	-	-	+	+	66.94767
52	+	+	-	-	+	+	28.70839
53	-	-	+	-	+	+	54.66564
54	+	-	+	-	+	+	28.71268
55	-	+	+	-	+	+	68.04705
56	+	+	+	-	+	+	29.16309
57	-	-	-	+	+	+	45.72794

Standard order	Pr[truth]	Tree depth	Number of alternatives	Alternative weight	Population	Number of answers	MAPE
58	+	-	-	+	+	+	4.47782
59	-	+	-	+	+	+	23.84796
60	+	+	-	+	+	+	13.90072
61	-	-	+	+	+	+	41.23229
62	-	+	+	+	+	+	25.70945
63	+	-	+	+	+	+	13.88817
64	+	+	+	+	+	+	14.41732

Table 6.7: MAPE measurements for the experiment using -0.5 and +0.5 as coded value inputs

Paper VI

Boel Nelson, Jenni Reuben

SoK: Chasing Accuracy and Privacy, and Catching Both in Differentially Private Histogram Publication

Transactions on Data Privacy 13:3 (2020) 201 - 245

SoK: Chasing Accuracy and Privacy, and Catching Both in Differentially Private Histogram Publication

Abstract

Histograms and synthetic data are of key importance in data analysis. However, researchers have shown that even aggregated data such as histograms, containing no obvious sensitive attributes, can result in privacy leakage. To enable data analysis, a strong notion of privacy is required to avoid risking unintended privacy violations.

Such a strong notion of privacy is *differential privacy*, a statistical notion of privacy that makes privacy leakage quantifiable. The caveat regarding differential privacy is that while it has strong guarantees for privacy, privacy comes at a cost of accuracy. Despite this trade-off being a central and important issue in the adoption of differential privacy, there exists a gap in the literature regarding providing an understanding of the trade-off and how to address it appropriately.

Through a systematic literature review (SLR), we investigate the state-of-the-art within accuracy improving differentially private algorithms for histogram and synthetic data publishing. Our contribution is two-fold: 1) we identify trends and connections in the contributions to the field of dif-

ferential privacy for histograms and synthetic data and 2) we provide an understanding of the privacy/accuracy trade-off challenge by crystallizing different dimensions to accuracy improvement. Accordingly, we position and visualize the ideas in relation to each other and external work, and deconstruct each algorithm to examine the building blocks separately with the aim of pinpointing which dimension of accuracy improvement each technique/approach is targeting. Hence, this systematization of knowledge (SoK) provides an understanding of in which dimensions and how accuracy improvement can be pursued without sacrificing privacy.

7.1 Introduction

Being able to draw analytical insights from data sets about individuals is a powerful skill, both in business, and in research. However, to enable data collection, and consequently data analysis, the individuals' privacy must not be violated. Some strategies [1, 2, 3] for privacy-preserving data analysis focus on sanitizing data, but such approaches require identifying sensitive attributes and also does not consider auxiliary information. As pointed out by Narayanan and Shmatikov [4], personally identifiable information has no technical meaning, and thus cannot be removed from data sets in a safe way. In addition to the difficulty in modeling the extent of additional information that an adversary may possess from public sources in such data sanitizing approaches, the privacy notion of such approaches is defined as the property of the data set. However, it is proved in [5] that for essentially any non-trivial algorithm, there exists auxiliary information that can enable a privacy breach that would not have been possible without the knowledge learned from the data analysis. Consequently, a strong notion of privacy is needed to avoid any potential privacy violations, while still enabling data analysis.

Such a strong notion of privacy is *differential privacy* [6] (Section 7.2), in which the privacy guarantee is defined as the property of the computations on the data set. Differential privacy is a privacy model that provides meaningful privacy

guarantees to individuals in the data sets by quantifying their privacy loss. This potential privacy loss, is guaranteed independently of the background information that an adversary may possess. The power of differential privacy lies in allowing an analyst to learn statistical correlations about a population, while not being able to infer information about any one individual. To this end, a differential private analysis may inject random noise to the results and these approximated results are then released to the analysts.

Differential privacy has spurred a flood of research in devising differentially private algorithms for various data analysis with varying utility guarantees. Given a general workflow of a differentially private analysis, which is illustrated in Figure 7.1, we have identified four *places* (labeled A, B, C and D) for exploring different possibilities to improve accuracy of differential private analyses.

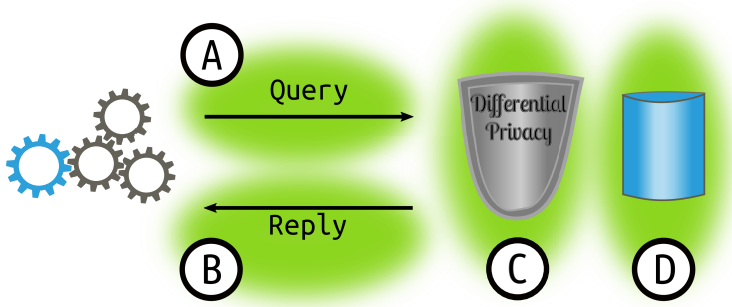


Figure 7.1: Places for accuracy improvement: A) Altering the query, B) Post-processing, C) Change in the release mechanism, D) Pre-processing.

In this work, we focus specifically on differentially private algorithms for histograms, and synthetic data publication. Histograms and synthetic data are particularly interesting because they both provide a way to represent summary of an underlying data set, thus may enable further analytical tasks executed over the summary of the data set. While, histograms represent a graphical summary

of frequency distribution of values of a specific domain in a data set, synthetic data is an approximate representation of data distribution of an underlying data set. Intrigued by the idea that there exists several ways to improve accuracy of privatized histograms and synthetic data without compromising privacy, we aim to systematically synthesize the state-of-the-art.

Advancement in research in differentially private histogram and synthetic data publication has received considerable interest within the computer science and statistics research communities [7, 8, 9]. However, only a few works systematically and critically assess the state-of-the-art differentially private, accuracy improving algorithms for releasing histograms or synthetic data. Li et al. [8] and Meng et al. [9] categorized different differentially private publication techniques for both histogram as well as synthetic data, and solely histograms respectively. However, their selection and categorization of the algorithms are not systematic. Further, the selected algorithms in their work are not exclusively accuracy improving techniques, but rather differentially private release mechanisms for histogram and synthetic data. That is, some of the surveyed algorithms do not boost the accuracy of an existing release mechanism by adding a modular idea, but instead invent new, monolithic algorithms. For example, some of the algorithms have discovered ways to release data that previously did not have a differentially private way of being released. Bowen and Liu [7], on the other hand, used simulation studies to evaluate several algorithms for publishing histograms and synthetic data under differential privacy. Their aim is quite different from ours, is to assess the accuracy[vii] and usefulness[viii] of the privatized results.

Consequently, to bridge the knowledge gap, the present paper aims to provide a systematization of knowledge concerning differentially private accuracy improving methods for histogram and synthetic data publication. To this end, we

[vii] We will use the terms accuracy and utility interchangeably when we refer to decreasing the error, i.e the distance between the privatized result and the true results.

[viii] We use the term usefulness to refer to the impact of the privatized results to conduct statistical inferences.

first review the main concepts related to differential privacy (Section 7.2), which are relevant to the qualitative analysis of state-of-the-art accuracy improving techniques for differentially private histogram and synthetic data publication (Section 7.5). However, before focusing on the qualitative analysis, we present our method to conduct a systematic review of literature that enable a methodological rigor to the results of the qualitative analysis (Section 7.3) and a review of general characteristics of the identified accuracy improving techniques (Section 7.4). We further study the composability of accuracy improvement techniques within the constraints of differential privacy in relation to the results of our analysis in order to pave the way for future research (Section 7.6). Overall, this systematization of knowledge provides a conceptual understanding of enhancing accuracy in the light of privacy constraints (Section 7.7).

Our main contributions are:

1. A technical summary of each algorithms in order to provide a consolidate view of the state-of-the-art (Section 7.4).
2. Categorization that synthesize the evolutionary relationships of the research domain in differential privacy for histogram and synthetic data publication (Section 7.5.1).
3. Categorization of the state-of-the-art, which is based on the conceptual relationships of the identified algorithms (Section 7.5.2).

7.2 Differential Privacy

Differential privacy [6] is a statistical definition that enables privacy loss to be quantified and bounded. In differential privacy, privacy loss is bounded by the parameter ε. To achieve trivial accuracy improvement, ε can be tweaked to a higher value, as this gives less privacy (greater privacy loss) which means more accuracy. In this paper we only consider accuracy improvements in settings where ε is fixed.

We formally define ε-differential privacy in Definition 1, based on Dwork [5]. The parameter ε is usually referred to as the *privacy budget*. Essentially, ε is the cost in terms of privacy loss for an individual participating in an analysis.

Definition 1 (ε-Differential Privacy)**.** *A randomized algorithm f' gives ε-differential privacy if for all data sets D_1 and D_2, where D_1 and D_2 are neighboring, and all $\mathcal{S} \subseteq$ Range(f'),*

$$Pr[f'(D_1) \in \mathcal{S}] \leq e^{\varepsilon} \times Pr[f'(D_2) \in \mathcal{S}]$$

A relaxed version of differential privacy is (ε, δ)-differential privacy Dwork et al. [10], which we define in Definition 2. (ε, δ)-differential privacy is primarily used to achieve better accuracy, but adds a subtle, probabilistic dimension of privacy loss. (ε, δ)-differential privacy is sometimes also called *approximate differential privacy* [11].

Definition 2 ((ε, δ)-Differential Privacy)**.** *A randomized algorithm f' is (ε, δ)-differentially private if for all data sets D_1 and D_2 differing on at most one element, and all $\mathcal{S} \subseteq$ Range(f'),*

$$Pr[f'(D_1) \in \mathcal{S}] \leq e^{\varepsilon} \times Pr[f'(D_2) \in \mathcal{S}] + \delta$$

Theoretically, in ε-differential privacy each output is *nearly* equally likely and hold for *any* run of algorithm f', whereas (ε, δ)-differential privacy for *each pair* of data sets (D_1, D_2) in extremely unlikely cases, will make some answer much less or much more likely to be released when the algorithm is run on D_1 as opposed to D_2 [12]. Still, (ε, δ)-differential privacy ensures that the absolute value of the privacy loss is bounded by ε with probability at least 1-δ [12]. That is, the probability of gaining significant information about one individual, even when possessing all other information in the data set, is at most δ.

To satisfy differential privacy, a randomized algorithm perturbs the query answers to obfuscate the impact caused by differing one element in the data set.

Such perturbation can for example be introduced by adding a randomly chosen number to a numerical answer. Essentially, the maximum difference *any* possible record in the data set can cause dictates the magnitude of noise needed to satisfy differential privacy. This difference is referred to as the algorithm's L_1 sensitivity, which we define in Definition 3, based on Dwork et al. [6].

Definition 3 (L_1 Sensitivity). *The L_1 sensitivity of a function $f : D^n \rightarrow \mathbb{R}^d$ is the smallest number Δf such that for all D_1, $D_2 \in D^n$ which differ in a single entry,*

$$\|f(D_1) - f(D_2)\|_1 \leq \Delta f$$

Since differential privacy is a property of the algorithm, as opposed to data, there exists many implementations of differentially private algorithms. Thus, we will not summarize all algorithms, but instead introduce two early algorithms that are common building blocks, namely: the Laplace mechanism [6] and the Exponential mechanism [13].

We define the Laplace mechanism in Definition 4, based on the definition given by Dwork [14]. The Laplace mechanism adds numerical noise, and the probability density function is centered around zero, meaning that noise with higher probability (than any other specific value) will be zero.

Definition 4 (Laplace mechanism). *For a query f on data set D, the differentially private version, f', adds Laplace noise to f proportional to the sensitivity of f:*

$$f'(D) = f(D) + Lap(\Delta f/\varepsilon)$$

Furthermore, we define the Exponential mechanism (EM) in Definition 5 based on the definition given by McSherry and Talwar [13]. The intuition behind EM is that the probability of not perturbing the answer is slightly higher than perturbing the answer. EM is particularly useful when Laplace does not make sense, for example when queries return categorical answers such as strings,

but can also be used for numerical answers. The reason EM is so flexible is that the utility function can be replaced to score closeness to suit the given domain.

Definition 5 (Exponential mechanism (EM))**.** *Given a utility function* $u : (D \times R) \rightarrow R$*, and a data set* D*, we define the differentially private version,* u'*:*

$$u'(D, u) = \Big\{ \textit{return } r, \textit{where } r \textit{ ranges over } R, \textit{ with probability } \propto exp\frac{\varepsilon u(D,r)}{2\Delta u} \Big\}$$

The semantic interpretation of the privacy guarantee of differential privacy rests on the definition of what it means for a pair of data sets to be neighbors. In the literature, the following two variations of neighbors are considered when defining differential privacy: unbounded and bounded.

Definition 6 (Unbounded Differential Privacy)**.** *Let* D_1 *and* D_2 *be two data sets where* D_1 *can be attained by adding or removing a single record in* D_2*. With this notion of neighbors, we say that we have unbounded differential privacy.*

Definition 7 (Bounded Differential Privacy)**.** *Let* D_1 *and* D_2 *be two data sets where* D_1 *can be attained by changing a single record in* D_2*. With this notion of neighbors, we say that we have bounded differential privacy.*

Distinguishing between the definition of neighboring data sets is important, because it affects the global sensitivity of a function. The sizes of the neighboring data sets are fixed in the bounded differential privacy definition whereas, there is no size restriction in the unbounded case.

In the case of graph data sets, a pair of graphs differ by their number of edges, or number of nodes. Therefore, there exists two variant definitions in literature [15] that formalize what it means for a pair of graphs to be neighbors. Nevertheless, these graph neighborhood definitions are defined only in the context of unbounded differential privacy.

Definition 8 (Node differential privacy [15]). *Graphs* $G = (V, E)$ *and* $G^{'} = (V^{'}, E^{'})$ *are* node-neighbors *if:*

$$V^{'} = V - v,$$
$$E^{'} = E - \{(v_1, v_2) \mid v_1 = v \vee v_2 = v\},$$

for some node $v \in V$.

Definition 9 (Edge differential privacy [15]). *Graphs* $G = (V, E)$ *and* $G^{'} = (V^{'}, E^{'})$ *are* edge-neighbors *if:*

$$V = V^{'},$$
$$E^{'} = E - \{e\},$$

for some edge $e \in E$.

In certain settings, ε grows too fast to guarantee a meaningful privacy protection. To cater for different applications, in particular in settings where data is gathered dynamically, different *privacy levels* have been introduced that essentially further changes the notion of neighboring data sets by defining neighbors for data streams. These privacy levels are, user level privacy [16], event level privacy [17], and w-event level privacy [18].

Definition 10 (User Level Privacy). *We say that a differentially private query gives user level privacy (pure differential privacy), when all occurrences of records produced by one user is either present or absent.*

Essentially, for user level privacy, all records connected to one individual user shares a joint privacy budget.

Definition 11 (Event Level Privacy). *We say that a differentially private query gives event level privacy, when all occurrences of records produced by one group of events, where the group size is one or larger, is either present or absent.*

With event level privacy, each data point used in the query can be considered independent and thus have their own budget.

Definition 12 (w-event Level Privacy)**.** *We say that a differentially private query gives w-event level privacy, when a set of w occurrences of records produced by some group of events, where the group size is one or larger, is either present or absent. When $w = 1$, w-event level privacy and event level privacy are the same.*

For w-event level privacy, w events share a joint privacy budget.

7.3 Method

We conducted a systematic literature review (SLR) [19] to synthesize the state-of-the-art accuracy improving techniques for publishing differentially private histograms as well as synthetic data. Systematic literature review, which, hereafter we will refer to as systematic review when describing generally, is a method to objectively evaluate all available research pertaining to a specific research question or research topic or phenomena of interest [19]. Although, the method is common in social science and medical science disciplines, the Evidence-Based Software Engineering initiative [20] have been influential in the recognition of systematic review as the method to integrate evidence concerning a research area, a research question or phenomena of interest in software engineering research. Systematic review provides methodological rigor to literature selection and synthesization as well as to the conclusion drawn as a result of the synthesization. The method consists of several stages that are grouped into three phases. The phases of systematic review are; i) planning the review, ii) conducting the review and iii) reporting the review.

Planning the review phase underpins the need for a systematic review concerning a research topic, a research question or phenomena of interest. Hence, in the planning stage, a review protocol that defines the research questions, as well as strategies for conducting the literature review is developed in order to minimize the likelihood of researcher bias in the selection of literature.

Following the specification of the search, the selection and the data synthesis strategy, the review is conducted (conducting the review phase) in an orderly manner. Thus, the first stage of the execution of a systematic review is the identification of all available literature. This stage involves the construction of search queries and identification of all relevant scholarly databases. After the identification of literature on a given topic of interest, they need to be evaluated for relevance, which usually is determined through a set of selection criteria. The selected literature for a systematic review is generally referred to as primary studies. Then, in order to synthesize the results of the primary studies, data are extracted from each primary study for the analysis that is the final stage of the conducting the review phase.

Reporting the review involves the documentation of the systematic review process and the communication of the results of the systematic review.

In the following subsections we describe in detail, the process we undertake in our SLR. Figure 7.2 shows the high-level view of the processes followed in our SLR.

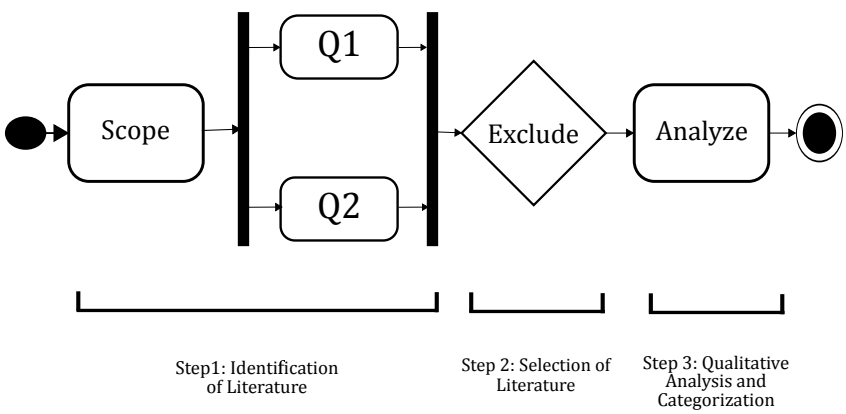


Figure 7.2: Workflow of processes followed in our SLR.

7.3.1 Identification of Literature

A thorough and unbiased search for literature is the essence of a SLR. In this SLR, we used a scholarly search engine, Microsoft Academic (MA) [21, 22], primarily for two reasons. First, for its semantic search functionality and second, for its coverage.

Semantic search leverages *entities* such as field of study, authors, journals, institutions, etc., associated with the papers. Consequently, there is no need to construct search strings with more keywords and synonyms, rather, a natural language query can be constructed with the help of search suggestions for relevant entities.

Secondly, regarding the coverage of MA. MA's predecessor, Microsoft Academic Search (MAS), suffered from poor coverage as pointed out by Harzing [23]. However, after relaunching MA its coverage has grown over the years [24, 25]. In 2017, Hug and Brändle [26] compared the coverage of MA to Scopus and Web of Science (WoS), and found that MA has higher coverage for book-related documents and conferences, and only falls behind Scopus in covering journal articles. More recently, in 2019, Harzing [27] compared the coverage of Crossref, Dimensions, Google Scholar (GS), MA, Scopus and WoS, and found that GS and MA are the most comprehensive free search engines. Accordingly, we have chosen to use MA since it has both adequate coverage and semantic search, while for example GS lacks semantic search.

We used two queries, one focusing on histograms and the other on synthetic data. The queries are as follows, with entities recognized by MA in bold text:

Q1: **Papers** about **differential privacy** and **histograms**
Q2: **Papers** about **differential privacy** and **synthetic data**

The search was performed on June 10 2019, and yielded 159 hits in total. 78

hits for **Q1** and 81 hits for **Q2**, which are examined for relevance in the next step of the SLR process.

7.3.2 Selection of Literature

We constructed and followed a set of exclusion criteria Table 7.1 in order to select the relevant literature that provides insights to our research aim. To reflect that we specifically wanted to focus on tangible, experimentally tested algorithms, we constructed the criteria to exclude papers that contribute to pure theoretical knowledge. To select papers, we examined the title and abstract of

Exclude if the paper is...
1) not concerning differential privacy, not concerning accuracy improvement, and not concerning histograms or synthetic data.
2) employing workflow actions, pre-processing/post-processing/algorithmic tricks but not solely to improve accuracy of histograms or synthetic data.
3) a trivial improvement to histogram or synthetic data accuracy through relaxations of differential privacy or adversarial models.
4) concerning local sensitivity as opposed to global sensitivity.
5) not releasing histograms/synthetic data.
6) pure theory, without empirical results.
7) about a patented entity.
8) a preprint or otherwise unpublished work.
9) not peer reviewed such as PhD thesis/master thesis/demo paper/poster/extended abstract.
10) not written in English.

Table 7.1: List of exclusion criteria followed in our SLR.

each paper against the exclusion criteria. When the abstract matches any one of the criteria, the paper is excluded, otherwise the paper is included. When it was unclear from the abstract that a contribution is empirical or pure theory, we

looked through the body of the paper to make our decision. In the course of this stage, in order to ensure the reliability of the decision concerning the inclusion of a literature in our systematic review, both the authors have independently carried out the selection of literature stage. When comparing the decisions of the authors, if there exist a disagreement, we discussed each disagreement in detail in relation of the criteria in Table 7.1 and resolved it. For the full list of excluded papers along with the reason for exclusion, see Section 7.A.

In the end, a total of 35 (after removing duplicates) papers were selected for the qualitative analysis.

7.3.3 Qualitative Analysis and Categorization

The most common framework found in the literature to analyse and understand a domain of interest, is classification schemes [28]. It concerns the grouping of objects with similar characteristics in a domain. Our aim is to synthesize; i) on the one hand, trends and relationships among each papers and ii) on the other hand, conceptual understanding of the privacy/accuracy trade-off in the differentially private histogram and synthetic data research. Therefore, from each paper we extracted distinct characteristics of the algorithms, evaluation details of the algorithms as well as design principles such as aim of the solution and motivation for using a particular technique. These characteristics are inductively analyzed for commonality, which follows, though not rigorously, the empirical-to-conceptual approach to taxonomy development defined by Nickerson et al. [28]. The categorization that resulted from the qualitative analysis are presented in Section 7.5.

Deviation from the systematic review guidelines in [19]: The review protocol for our SLR is not documented in the planning stage as specified by the original guidelines but rather documented in the reporting the review stage. This is largely due to the defined focus of our SLR, which is on the privacy/accuracy trade-off associated with differentially private algorithms for publishing

histograms and synthetic data. Hence, the search strategy and selection criteria do not call for an iteration and an account of the changes in the process. Further, in our SLR we do not consider a separate quality assessment checklist as prescribed by the SLR guidelines. However, in our SLR the quality of the primary studies is ensured through our detailed selection criteria that involves objective quality assessment criteria for example the criterion to include only peer-reviewed scientific publications in the SLR. Furthermore, the quality of the results of our SLR is ensured through the exclusion of some of the selected primary studies because the algorithms in those studies lack comparable properties in order to perform a fair comparison with other selected algorithms. Additionally, during the analysis we surveyed additional relevant literature from the related work sections of the primary studies, which adds to the quality of the results of our SLR.

7.4 Overview of Papers

After analyzing the 35 included papers, 27 papers [29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55] where found to be relevant. All included papers and their corresponding algorithms are listed in the ledger in Table 7.2. We illustrate the chronological publishing order of the algorithms, but note that within each year, the algorithms are sorted on the first author's last name, and not necessarily order of publication.

2010	Boost	Hay et al. [29]
2011	PMost, BMax	Ding et al. [30]
	Privelet, Privelet^{+}, Privelet*	Xiao et al. [56, 31]
2012	EFPA, P-HP	Acs et al. [32]
2013	NF, SF	Xu et al. [57, 33]
2014	DPCopula	Li et al. [34]
	C*i*TM	Lu et al. [35]
	PeGS, PeGS.rs	Park et al. [58], Park and Ghosh [36]
	DPCube	Xiao et al. [59, 60, 37]
	PrivBayes	Zhang et al. [38]
	AHP	Zhang et al. [39]
2015	RG	Chen et al. [40]
	ADMM	Lee et al. [41]
	DSAT, DSFT	Li et al. [42]
2016	(θ,Ω)-Histogram, θ-CumHisto	Day et al. [43]
	BPM	Wang et al. [44]
	PrivTree	Zhang et al. [45]
2017	DPCocGen	Benkhelif et al. [46]
	SORTaki	Doudalis and Mehrotra [47]
	Pythia, Delphi	Kotsogiannis et al. [48]
	Tru, Min, Opt	Wang et al. [49]
	DPPro	Xu et al. [50]
2018	T$^{\lambda}$	Ding et al. [51]
	GGA	Gao and Ma [52]
	PriSH	Ghane et al. [53]
2019	IHP, mIHP	Li et al. [61, 54]
	RCF	Nie et al. [55]

Table 7.2: Chronological ledger for the papers. Note that the abbreviation 'ADMM' is due to Boyd et al. [62], whereas Lee et al. [41]'s work is an extension that uses the same abbreviation.

Beware that some algorithms, for example NF, SF, have appeared in publications twice, first in a conference paper and then in an extended journal version. When a paper has two versions, we will refer to the latest version in our comparisons, but we include all references in the paper ledger for completeness. Furthermore, eight papers were excluded based on our qualitative analysis. Each decision is motivated in Section 7.6.2, and those eight papers hence do not appear in the paper ledger.

Furthermore, in Tables 7.3 and 7.4, we present objective parameters regarding the settings around the algorithms in each paper, for example the characteristics of the input data they operate on, and the metric used to measure errors. Our intention is that this table will allow for further understanding of which algorithms are applicable given a certain setting when one searches for an appropriate algorithm, but also to understand which algorithms are directly comparable in the scope of this SLR.

	Key	Meaning
Data	$\bowtie$	Correlated
	$\vec{x}$	Dynamic
	$\odot$	Sparse
	$\bar{x}$	Static
Dimension	$*$	Multi
	1D	Single
Mechanism	EM	Exponential mechanism
	LAP	Laplace mechanism
	MM	Matrix mechanism
	RR	Randomized response

	Key	Meaning
Metric	AVD	Average Variation Distance
	KS	Kolmogorov-Smirnov distance
	KL	Kullback-Leibler divergence
	$\ell 1$	L1 distance
	$\ell 2$	L2 distance
	MAE	Mean absolute error
	MISS	Misclassification rate
	MPE	Mean percentage error
	MSE	Mean squared error
	NWSE	Normalized weighted square error
	SAQ	Scaled average per query
Relation	♣	Bounded
	♢	Unbounded

Table 7.3: Meaning of symbols and abbreviations.

Reference	Definition	Level	Relation	Dimension	Input	Mechanism	Metric	Output
[29]	ε	?	♢	1D	$\bar{x}$	Lap	MAE	Histogram
[30]	ε	?	♢	*	$\bar{x}$, $\bowtie$	Lap	MAE	Cuboids
[31]	ε	?	♣	1D, *	$\bar{x}$	Lap	MAE, MPE	Range count queries
[32]	ε	?	♢	1D	$\bar{x}$, $\bowtie$	Lap, EM	KL, MSE	Histogram
[33]	ε	?	♢	1D	$\bar{x}$	Lap, EM	MAE, MSE	Histogram
[34]	ε	?	♢	*, $\odot$	$\bar{x}$	Lap	MAE, MPE	Synthetic data

Reference	Definition	Level	Relation	Dimension	Input	Mechanism	Metric	Output
[35]	(ε,δ)	Entity	♢	*	$\bar{x}$, $\bowtie$	MM, Agnostic	MPE	Model
[36]	ε	?	♢	*	$\bar{x}$	Dirichlet prior	Rank corr.	Model
[37]	ε	?	♢	*	$\bar{x}$	Lap	MAE	Histogram
[38]	ε	?	♣	*, $\odot$	$\bar{x}$	Lap, EM	AVD, Miss	Synthetic data
[39]	ε	?	♢	1D	$\bar{x}$	Lap	KL, MSE	Histogram
[40]	ε	Event	♢	1D	$\vec{x}$, $\bowtie$	Lap	MSE	Histogram
[41]	ε	?	♢	*	$\bar{x}$	Lap, MM	MSE	Contingency table, Histogram
[42]	ε	User, w-event	♢	1D	$\vec{x}$, $\bowtie$	Lap	MAE, MPE	Histogram
[43]	ε	?	Node	*	$\bar{x}$	EM	KS, $\ell 1$	Histogram
[44]	ε	?	♣	1D	$\bar{x}$	RR	NWSE	Histogram
[45]	ε	?	♢	*	$\bar{x}$, $\bowtie$	Lap	MPE	Quadtree
[46]	ε	?	♢	*	$\bar{x}$, $\odot$	Lap	Hellinger	Partitioning
[47]	ε	?	♢	1D	$\bar{x}$	Lap	SAQ	Histogram
[48]	ε	?	♢	1D, *	$\bar{x}$	Lap, Agnostic	$\ell 2$, Regret	N/A
[49]	ε	?	♢	1D	$\bar{x}$, $\bowtie$	Lap	MSE	Histogram
[50]	(ε,δ)	?	♣	*	$\bar{x}$	Gaussian, MM	Miss, MSE	Matrix
[51]	ε	?	Node	*	$\bar{x}$	Lap	KS, $\ell 1$	Histogram
[52]	ε	?	♢	1D	$\vec{x}$	Lap	MAE	Histogram
[53]	ε	?	♢	*, $\odot$	$\bar{x}$, $\bowtie$	MWEM	KL, $\ell 1$	Histogram

Reference	Definition	Level	Relation	Dimension	Input	Mechanism	Metric	Output
[54]	ε	?	$\diamondsuit$	1D,*	$\bar{x}$, $\odot$	Lap, EM	KL, MSE	Histogram
[55]	ε	?	$\clubsuit$	1D	$\bar{x}$	RR	MSE	Histogram

Table 7.4: Mapping between papers to corresponding differential privacy definition, privacy level, neighbor relationship, dimension of data, input data, use of mechanism, error metric and output data. Abbreviations and the corresponding symbols are explained in a separate table.

Note that the privacy level (user, event or w-event) was not explicitly stated in most papers, in which case we have attributed the privacy level as '?'. A '?' privacy level does not imply that the algorithm does not have a particular privacy level goal, but rather, that the authors did not explicitly describe what level they are aiming for. With this notice, we want to warn the reader to be cautious when comparing the experimental accuracy of two algorithms unless they in fact assume the same privacy level. For example, comparing the same algorithm but with either user level or event level privacy would make the event level privacy version appear to be better, whereas in reality it trivially achieves better accuracy through relaxed privacy guarantees.

In general, user level privacy tends to be the base case, as this is the level assumed in *pure* differential privacy [16], but to avoid making incorrect assumptions, we chose to use the '?' label when a paper does not explicitly state their privacy level.

Histogram	Hybrid	Synthetic Data
Hay et al. [29]	Lu et al. [35]	Li et al. [34]
Xiao et al. [31]	Ding et al. [30]	Park and Ghosh [36]

Acs et al. [32]	Xiao et al. [37]	Zhang et al. [38]
Xu et al. [33]	Lee et al. [41]	Xu et al. [50]
Zhang et al. [39]	Zhang et al. [45]	
Chen et al. [40]	Benkhelif et al. [46]	
Li et al. [42]	Kotsogiannis et al. [48]	
Day et al. [43]	Wang et al. [49]	
Wang et al. [44]	Li et al. [54]	
Doudalis and Mehrotra [47]		
Ding et al. [51]		
Gao and Ma [52]		
Ghane et al. [53]		
Nie et al. [55]		

Table 7.5: The papers grouped by their type of output, where hybrid internally uses histogram structures where synthetic data is sampled from.

Given that our two queries were designed to capture algorithms that either output synthetic data or a histogram, we examine the similarity between the strategies used in each algorithm. To this end, we manually represent the similarity between the algorithms' strategies based on their output in Table 7.5. We distinguish between the two kinds of outputs by their different goals: for histograms, the goal is to release *one optimal histogram* for a given query, whereas for synthetic data the goal is to release a data set that is optimized for some *given set of queries*. Some algorithms use similar approaches to the algorithms from the other query; and therefore we label them as hybrid. An example of a hybrid paper is Li et al. [54], since they both deal with one-dimensional histograms (IHP), and then re-use that strategy when producing multi-dimensional histograms (mIHP) that resembles the outputs of synthetic data papers.

7.5 Analysis

We present our qualitative analysis on 27 included papers from two different perspectives in the light of research in differential privacy histogram and synthetic data. First, from a evolutionary perspective for identifying trends and to position each contribution in the history of its research (Section 7.5.1). Second, from a conceptual perspective for understanding the trade-off challenge in the privacy and utility relationship (Section 7.5.2).

7.5.1 Positioning

In order to provide context, we studied *where* the algorithms originated from, and how they are connected to each other. To also understand *when* to use each algorithm, and which ones are comparable in the sense that they can be used for the same kind of analysis, we also investigate which algorithms are compared experimentally in the papers.

First, we explored and mapped out the relationships between the included algorithms. To further paint the picture of the landscape of algorithms, we analyzed the related work sections to find external work connected to the papers included in our SLR. We present our findings as a family tree of algorithms in Figure 7.3, which addresses from where they came.

Since our exploration of each algorithms' origin discovered papers outside of the SLR's queries, we also provide a ledger for (Table 7.6) *external* papers. When the authors had not designated a name for their algorithms, we use the abbreviation of the first letter of all author's last name and the publication year instead. Note that we have not recursively investigated the external papers' origin, so external papers are not fully connected in the family tree.

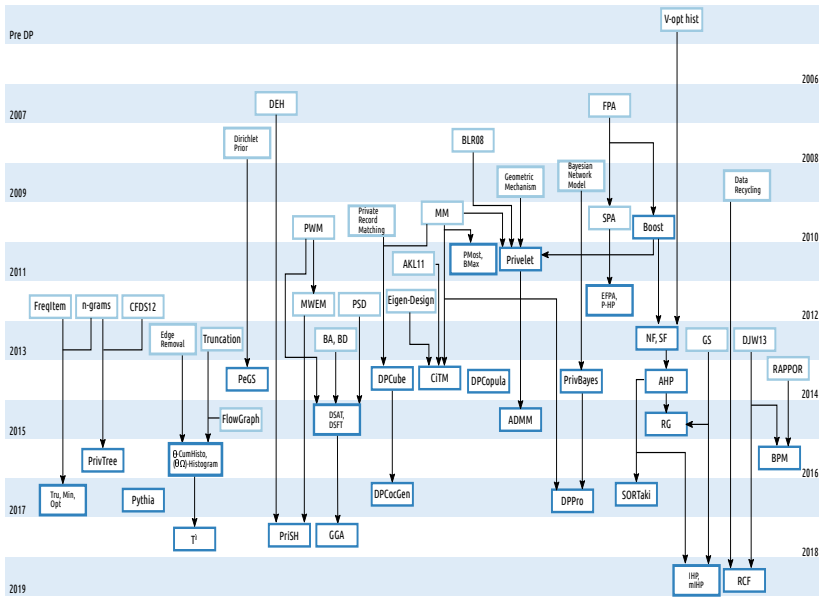


Figure 7.3: The family tree of algorithms. Light blue indicate papers not covered by the SLR, and the darker blue represents included papers.

Label	Author
AKL11	Arasu et al. [63]
Bayesian Network Model	Koller and Friedman [64]
BLR08	Blum et al. [65]
Budget Absorption (BA), Budget Distribution (BD)	Kellaris et al. [18]
CFDS12	Chen et al. [66]
Data Recycling	Xiao et al. [67]
Dirichlet Prior	Machanavajjhala et al. [68]
Distributed Euler Histograms (DEH)	Xie et al. [69]
DJW13	Duchi et al. [70]

Label	Author
Edge Removal	Blocki et al. [71]
Eigen-Design	Li and Miklau [72]
FlowGraph	Raskhodnikova and Smith [73]
Fourier Perturbation Algorithm (FPA)	Barak et al. [74]
FreqItem	Zeng et al. [75]
Geometric Mechanism	Ghosh et al. [76]
Grouping and Smoothing (GS)	Kellaris and Papadopoulos [77]
Matrix Mechanism (MM)	Li et al. [78]
MWEM	Hardt et al. [79]
n-grams	Chen et al. [80]
Private Multiplicative Weights (PMW)	Hardt and Rothblum [81]
Private Record Matching	Inan et al. [82]
Private Spatial Decompositions (PSD)	Cormode et al. [83]
RAPPOR	Erlingsson et al. [84]
Sampling Perturbation Algorithm (SPA)	Rastogi and Nath [85]
Truncation	Kasiviswanathan et al. [86]
V-opt hist	Jagadish et al. [87]

Table 7.6: Ledger for papers outside of the SLR.

From the family tree, we notice that there are several different lines of research present. One frequently followed line of research is that started by Xu et al. [33], NF, SF, which addresses the issue of finding an appropriate histogram structure (i.e. bin sizes) by creating a differentially private version of a v-optimal histogram. Essentially, EM is used to determine the histogram structure, and then the Laplace mechanism is used to release the bin counts. The idea by Xu et al. [33] is followed by AHP, RG, SORTaki, IHP and mIHP.

The matrix mechanism (MM) is a building block that is used in PMost, BMax, C*i*TM and DPPro. Apart from using the same release mechanism, they do not share many similarities as also becomes apparent when comparing their exper-

imental evaluation.

Only Pythia and DPCopula appears as orphaned nodes in the family tree. Pythia is special in the sense that it is not a standalone algorithm, but rather provides a differentially private way of choosing the 'best' algorithm for a given data set. DPCopula has a mathematical background in copula functions, which are functions that describe the dependence between multivariate variables. This approach of using copula functions is not encountered in any of the other papers.

To further put the algorithms into perspective, we explored which algorithms were used in their experimental comparisons. The comprehensive matrix of which algorithms are experimentally compared to each other in Table 7.7. This complements the fact table (Table 7.4) in addressing the question of when to use an algorithm, as algorithms that are compared experimentally can be used interchangeably for the same analysis. E.g, when NF is used, it can be swapped with for example IHP.

Algorithm	**Internal Comparison**	**External Comparison**
Boost	-	-
PMost, BMax	-	-
Privelet, Privelet^{+}, Privelet*	-	-
EFPA, P-HP	Boost, Privelet,NF, SF	SPA [85], MWEM [79]
NF, SF	Boost, Privelet	-
DPCopula	Privelet^{+}, P-HP	FP [88], PSD [83]
C*i*TM	-	-
PeGS, PeGS.rs	-	-
DPCube	Boost	Private Interactive ID3 [89], Private Record Matching [82]

PrivBayes	-	FPA[74], PrivGene [90], ERM [91]
AHP	NF, SF, P-HP	GS [77]
RG	-	BA [18], FAST [92]
ADMM	Boost, EFPA, P-HP, Privelet	LMM [78], RM [93]
DSAT, DSFT	-	-
(θ,Ω)-Histogram, θ-CumHisto	-	EdgeRemoval [71], Truncation [86], FlowGraph [73]
BPM	-	EM [13], Binary RR [70, 84]
PrivTree	Privelet*	UG [94, 95, 96], AG [94], Hierarchy [95], DAWA [97]
DPCocGen	PrivBayes	-
SORTaki	-	-
Pythia, Delphi	-	-
Tru, Min, Opt	Boost	n-grams [80], FreqItem [75],GS, DAWA, DPT [98]
DPPro	-	Private SVM [99], PriView [100], JTree [101]
T^{λ}	-	-
GGA	DSAT	-
PriSH	-	MWEM, DAWA
IHP, mIHP	Boost,EFPA, P-HP, SF, AHP	PSD, GS
RCF	Boost, NF	SHP [102]

Table 7.7: Algorithms used in empirical comparisons, divided by internal (included in the SLR) and external (excluded from the SLR) algorithms, sorted by year of publication. Comparisons with the Laplace mechanism and the author's own defined baselines (such as optimal) have been excluded from the table.

7.5.2 Categorization of Differentially Private Accuracy Improving Techniques

We observe from the algorithms in the 27 papers, there are three different dimensions to accuracy improvement in the context of differential privacy: **i)** *total noise reduction*, **ii)** *sensitivity reduction* and **iii)** *dimensionality reduction*.

i) **Total Noise Reduction** On the one hand, a histogram is published as statistical representation of a given data set (Goal I). On the other hand, histograms are published as a way to approximate the underlying distribution, which is then used to answer queries on the data set (Goal II). We refer to the latter as universal histograms: terminology adapted from [29]. In this dimension, optimizing the noisy end result (i.e differentially private histograms) provides opportunities for accuracy improvement.
ii) **Sensitivity Reduction** The global sensitivity of histogram queries is not small for graph data sets. Because, even a *relatively* small change in the network structure results in big change in the query answer. The accuracy improvement in this dimension follow from global sensitivity optimization.
iii) **Dimensionality Reduction** Publishing synthetic version of an entire data set consists of building a private statistical model from the original data set and then sampling data points from the model. In this dimension, inferring the underlying data distribution from a smaller set of attributes provides opportunities for accuracy improvement.

Dimension: Total Noise Reduction

In Table 7.8, we summarize the distinct techniques/approaches of the state-of-the-art from the point of view of reducing the total noise.

Category	Technique/Approach	Algorithms	Notes
Clustering	Bi-partite	BPM	
	Bisection	P-HP	
	Bisection	IHP, mIHP	
	MODL co-clustering [103]	DPCocGen	
	Matrix decomposition	Privelet^{+}	
	Weighted combination	AC	Least Square Method
	Retroactive Grouping	RG	Thresholded
	Selecting Top k	EFPA	
		CiTM	Key/foreign-key Relationships
		Min	Query Overlap
		SF	V-optimality
		NF	
		AHP	V-optimality
		(θ,Ω)-Histogram	V-optimality
		T$^{\lambda}$	Equi-width

Category	Technique/Approach	Algorithms	Notes
Consistency Check	Frequency Calibration	θ-CumHisto	Monotonicity Property
	Hierarchical Consistency	Opt	
	Least Square Minimization	Boost	
	Least Square Minimization	DPCube	
	Realizable model	CiTM	Linear-time Approximation
		PMost	Least Norm Problem
Hierarchical Decomposition	Binary Tree	Boost	
	kd-tree	DPCube	V-optimality
	Quadtree	PrivTree	
	Query Tree	CiTM	Correlation of i-Table Model
	Sequential Partitions	mIHP	t-value
Learning True Distribution	Reallocate Values	θ-CumHisto	Linear Regression, Powerlaw & Uniform distributions
	Rescaling Weights	PriSH	Query Absolute Error, Dependency Constraints

Category	Technique/Approach	Algorithms	Notes
Privacy Budget Optimization	Composition rule-based	C*i*TM	
	Threshold-driven Release	DSAT, DSFT	Adaptive-distance Qualifier, Fixed-distance Qualifier
	Threshold-driven Release	GGA	Fixed-distance Qualifier
	Weighted	BPM	
Sampling	Bernoulli Sampling	RG	
	Data Recycling	DRPP	
Sorting		AHP	
Transformation	Wavelet Transform	Privelet	
	Fourier Transformation	EFPA	
Threshold	Qualifying Weight	PMost	
	Qualifying Source-of-noise	BMax	
	Qualifying Source-of-noise	Tru	
	Sanitization	AHP	
	Wavelet Thresholding	Privelet*	

Table 7.8: Categorization of techniques/approaches used by each algorithm for total noise reduction. Additional qualifiers of each techniques are captured as notes.

▷ **Goal I:** When the goal is to publish some statistical summary of a given data set as a differentially private histogram, histogram partitions play an essential role in improving the accuracy of the end result. A histogram partitioned into finer bins reduces approximation error[ix] of the result, be-

[ix] Error caused by approximating the underlying distribution of data into histogram bins: intervals covering the range of domain values.

cause each data point is correctly represented by a bin. However, the Laplace mechanism for histograms adds noise of scale $\Delta f/\varepsilon$ to each histogram bin. In other words, a histogram that is structured to minimize the approximation error, would suffer more noise in order to satisfy differential privacy.

The most common approach to enhance the utility for this goal, is to identify optimal histogram partitions for the given data.

Algorithms P-HP, SF and (θ,Ω)-Histogram use the Exponential mechanism to find V-optimal histogram [87] partitions. However, the quality of the partitions drops as the privacy budget available for iterating the Exponential mechanism decreases. Hence, algorithms NF, AHP, DPCocGen instead operate on the non-optimized noisy histogram for identifying sub-optimal partitions for the final histogram. To further improve the quality of the partitions that are based on the non-optimized noisy histogram, in AHPsorting technique is used.

For the same goal described above, if the given data are bitmap strings then one opportunity for accuracy improvement is to vary the amount of noise for various histogram bins. Algorithm BPM uses a bi-partite cut approach to partition a weighted histogram into bins with high average weight and bins with low relative weight. Further, in BPM the privacy budget ε is carefully split between the bins such that the heavy hitters, i.e. bins with high count, enjoy less noise. Algorithm AC uses weighted combination approach in terms of least square method in order to find optimal histogram partitions. Sample expansion through recycling the data points is another interesting approach for enhancing the accuracy of histograms over bitmap strings.

In the case of dynamic data sets, it is desirable to sequentially release the statistical summary of evolving data set at a given point in time. The most common approach is to limit the release of histograms, when there is a change in the data set for avoiding early depletion of privacy budget. Algorithms DSFT, DSAT and GGA uses distance-based sampling to monitor significant updates to the input

data set. In algorithm RG an adaptive sampling process uses Bernoulli sampling for change detection in the data set. Further, in RG a novel histogram partitioning approach called retroactive grouping is introduced to enhance the accuracy of the end result.

- ▷ **Goal II:** When the histograms are used to answer workload of allowable queries. Laplace noise accumulates (sequential composition) as the number of queried histogram bins increases in order to answer the workload (covering large ranges of domain values). However, if the answer to the workload can be constructed by finding a linear combination of fewer bins, then the accuracy of the final answer will be significantly improved.

Algorithms Boost, DPCube, PrivTree, C*i*TM and mIHP employ an approach, where the domain ranges are hierarchically structured, typically in a tree structure. The intuition is, to find the fewest number of internal nodes such that the union of these ranges equals the desired range in the workload. To further improve the accuracy in the context of sequential composition, algorithm C*i*TM uses composition rule-based privacy budget optimization. Transformation techniques such as wavelet transform (Privelet) and Fourier transform (EFPA) are also used to model linear combination of domain ranges.

Another approach to reduce the accumulate noise in the context of universal histograms is to contain the total noise below a threshold. In BMax the maximum noise variance of the end result is contained within a threshold.

Furthermore, constraints are imposed in the output space of possible answers, which are then verified in the post-processing step to identify more accurate answers in the output space.

Preserving the dependency constraint is important for answering range queries over spatial histograms. To this end, in algorithm PriSH, true distribution of the underlying data set is learned from private answers to carefully chosen informative queries. Separately, to estimate the tail distribution of the final noisy histogram, algorithm θ-CumHisto uses some prior distribution to reallocate count

values.

Dimension: Sensitivity Reduction

In Table 7.9, we summarize the distinct techniques/approaches of the state-of-the-art from the point of view of reducing the global sensitivity.

Category	Technique/Approach	Algorithms	Notes
Neighbor Relation	Redefine	C*i*TM	Propagation Constraints
Projection	Edge Addition	(θ,Ω)-Histogram θ-CumHisto	Network Degree Bounded
	Edge Deletion	T^{λ}	Mutual Connections Bounded

Table 7.9: Categorization of techniques/approaches used by each algorithms for sensitivity reduction. Additional qualifiers of each techniques are captured as notes.

In graph data sets, global sensitivity becomes unbounded, for example, change in a node and its edges, in the worst case affects the whole structure (i.e involving all the nodes) of the network under *node differential privacy*. Bounding the network degree is one of the common approaches for containing the global sensitivity for analysis under *node differential privacy*. Techniques, edge addition ((θ,Ω)-Histogram, θ-CumHisto) and edge deletion (T^{λ}) are used to bound the size of the graph. Consequently, the noise required to satisfy *node differential privacy* will be reduced.

When there exists no *standard* neighborhood definition for the differential privacy guarantee in the light of correlated data structures. In the C*i*TM algorithm that operates on relational databases with multiple relation correlations, the neighbor relation is redefined.

Dimension: Dimensionality Reduction

In Table 7.10, we summarize the distinct techniques/approaches of the state-of-the-art from the point of view of reducing the data dimensions.

Category	Technique/Approach	Algorithms	Notes
Consistency check	Eigenvalue Procedure [104]	DPCopula	
Projection	Hashing Trick [105]	PeGS	
Privacy Budget Optimization	Reset-then-sample	PeGS.rs	
Transformation	Bayesian Network	PrivBayes	
	Copula Functions	DPCopula	
	Random Projection	DPPro	Johnson-Lindenstrauss Lemma

Table 7.10: Categorization of techniques/approaches used by each algorithm for data dimensionality reduction. Additional qualifiers of each techniques are captured as notes.

The most common approach to accuracy improvement in this dimension is to build statistical models that approximate the full dimensional distribution of the data set from multiple set marginal distributions. Some of techniques to approximate joint distribution of a data set are Bayesian Network (PrivBayes) and Copula functions (DPCopula). Furthermore, projection techniques from high-dimensional space to low-dimensional sub-spaces are shown to improve accuracy as less noise is required to make the smaller set of low-dimensional sub-spaces differentially private. Projection techniques found in the literature are, feature hashing using the hashing trick (PeGS) and random projection based on the Johnson-Lindenstrauss Lemma (DPPro).

In DPCopula, eigenvalue procedure is used in the post-processing stage to achieve additional gain in accuracy. Unexpectedly, reset-then-sample approach grouped under privacy budget optimization algorithmic category appear in this dimension, because the PeGS.rs algorithm supports multiple synthetic data set instances.

Summary

Figure 7.4 summarizes the categorization of differentially private accuracy improving techniques. Techniques identified in each accuracy improving dimensions are grouped into specific categories. The algorithmic categories are further partially sub-divided by the input data they support. Query answer relates to the type of release rather than to the input data, but the assumption is that the other mentioned data types, they implicitly specify the type of release.

The further the algorithmic category is located from the center of the circle, the more common is that category in that particular accuracy improvement dimension. Subsequently, clustering is the most commonly employed category for the total noise reduction dimension. Interestingly, same set of categories of accuracy improving algorithms are employed for dynamic data and bitmap strings, in the context of total noise reduction dimension. Hierarchical decomposition, consistency check and learning true distribution are primarily used in the context of releasing a histogram for answering workload of queries. It should be noted that the consistency check technique is used in the dimensionality reduction dimension as well but the usage of the technique is conditional.

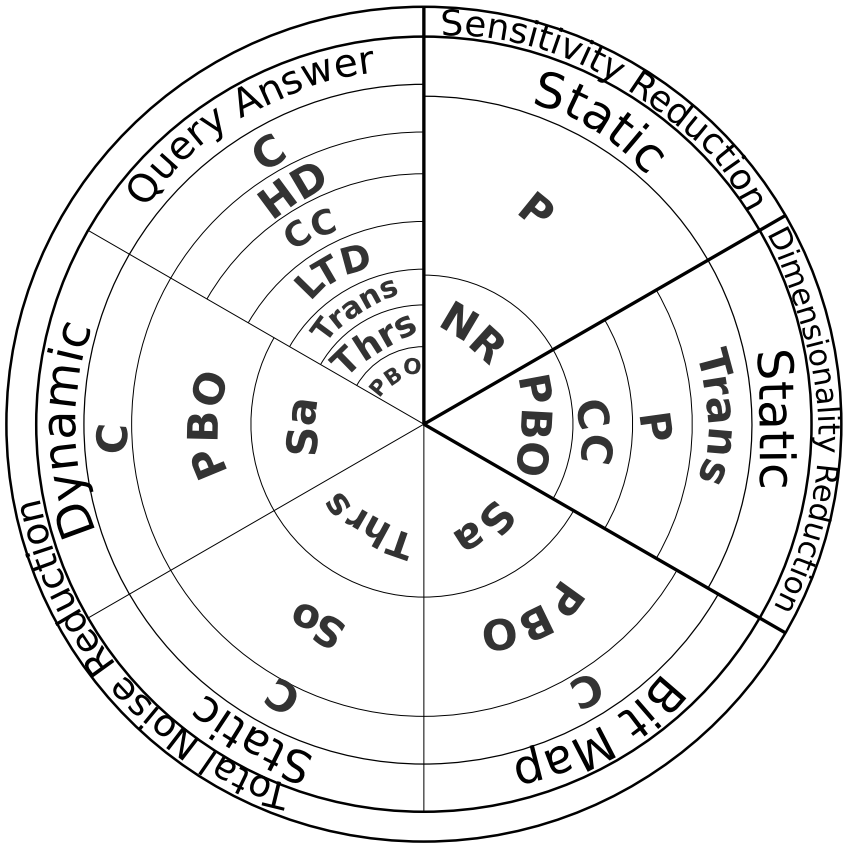


Figure 7.4: Conceptualization of accuracy improving techniques in the context of differential privacy: Abbreviations: **C**: Clustering, **CC**: Consistency Check, **HD**: Hierarchical Decomposition, **LTD**: Learning True Distribution, **NR**: Neighborhood Redefine, **P**: Projection, **PBO**: Privacy Budget Optimization, **Thrs**: Threshold, **Trans**: Transformation, **Sa**: Sampling, **So**: Sorting.

7.6 Discussion and Open Challenges

One limitation of this paper is that the scope of our SLR is limited to papers with empirical results. We have chosen empirical measurement of accuracy, since it can provide a less pessimistic understanding of error bounds, as opposed to analytical bounds. However, in our analysis (Section 7.5) of the papers, we studied related theoretical aspects of accuracy improvements and put the surveyed papers into context by tracing their origin, illustrated in Figure 7.3. As such, we can guide the interested reader in the right direction, but we do not provide an analysis of theoretical results.

Next (Section 7.6.1), we identify possible future work, mainly related to composability of the different techniques. It is not clear exactly which techniques compose, or how many techniques from each place that can be used to achieve accuracy improvements. Hence, open challenges include both coming up with new accuracy techniques for each place as well as combining techniques in meaningful, composable ways. Last Section 7.6.2, we list the papers that were excluded as part of our qualitative analysis.

7.6.1 Composability of Categories

From the dimensions identified in our analysis, we continue by investigating how techniques from different categories *may* be composed. We also connect the papers with the *place* [x] their algorithm operates on in Table 7.11.

	A	B	C	D
Hay et al. [29]		✓		
Ding et al. [30]		✓	✓	
Xiao et al. [31]				✓
Acs et al. [32]			✓	✓

[x] Places refers to different points in the workflow of a typical differentially private analysis, see Figure 7.1

	A	B	C	D
Xu et al. [33]			✓	
Li et al. [34]			✓	
Lu et al. [35]		✓		✓
Park and Ghosh [36]		✓		✓
Xiao et al. [37]			✓	
Zhang et al. [38]			✓	
Zhang et al. [39]			✓	
Chen et al. [40]		✓		
Lee et al. [41]		✓		
Li et al. [42]	✓		✓	
Day et al. [43]	✓	✓	✓	
Wang et al. [44]			✓	
Zhang et al. [45]			✓	
Benkhelif et al. [46]		✓	✓	
Doudalis and Mehrotra [47]			✓	
Kotsogiannis et al. [48]	✓	✓		✓
Wang et al. [49]			✓	
Xu et al. [50]			✓	
Ding et al. [51]	✓			✓
Gao and Ma [52]			✓	
Ghane et al. [53]		✓	✓	
Li et al. [54]			✓	
Nie et al. [55]		✓		✓

Table 7.11: Mapping the papers to each place where: A) Altering the query, B) Post-processing, C) Change in mechanism, D) Pre-processing.

We believe a technique from one place is possible to compose with techniques from another place, since the places are designed to be a sequential representation of the data analysis. An open challenge derived from Table 7.11 is boosting each algorithm's accuracy by adding more techniques, either in a place which

does not yet have any accuracy improvement, or together with the already existing techniques. For example, an algorithm that has improvement in place B (post-processing) may be combined with place A, C and/or D. Similarly, it may be possible to compose one technique from place B with another technique also from place B.

Next, we will illustrate how composability is already achieved by giving a few examples of how techniques are composed in the included papers.

Place A: Altering the Query

Altering the query targets *sensitivity reduction*, as sensitivity is a property of the query. Our take away from the SLR is that there are mainly two tricks to altering the query:

1. When an analysis requires a high sensitivity query, replace the query with an approximate query, or break the query down into two or more sub-queries.
2. Use sampling to avoid prematurely exhausting the privacy budget.

Item 1: For example, a histogram query is broken down into two separate queries: a clustering technique based on the exponential mechanism and usually a Laplace counting query, as in the case with Xu et al. [33] and consecutive work.

By breaking down the query, the sensitivity reduction can increase accuracy, but it needs to be balanced against the source of accumulated noise that is introduced by multiple queries. In particular, when breaking down a query, the privacy budget needs to be appropriately distributed between the sub-queries. For example, when breaking a histogram into a clustering query and then a count query, one could choose to give more budget to the clustering step to find a tighter histogram structure, but that would come at the cost of less accuracy for the count query.

Item 2: When an analysis is done on dynamic data, it is possible to unintentionally include the same data points in multiple queries, and ending up 'paying' for them multiple times. Li et al. [42] mitigates this source of accumulated noise by deploying sampling. It is also possible to use sampling for static data, for example, Delphi by Kotsogiannis et al. [48] could be trained on a sample of the full data set, if no public training data is available.

Place B: Post-processing

Post-processing targets *total noise reduction*, usually by exploiting consistency checks or other known constraints. Since post-processing is done on data that has been released by a differentially private algorithm, post-processing can always be done without increasing the privacy loss. However, post-processing can still decrease accuracy if used carelessly. In our SLR, the main post-processing idea is:

1. Finding approximate solutions to get rid of inconsistencies through *constrained inference* [29].
2. Applying consistency checks that would hold for the raw data.

Item 1: Boost is already being combined with several algorithms that release histograms, for example NF and SF. ADMM is a similar, but more generic solution that has been applied to more output types than just histograms. In fact, Lee et al. [41] claims ADMM can re-use algorithms use for least square minimization, which means Boost should be possible to incorporate in ADMM. Consequently, we believe ADMM would compose with most algorithms due to its generic nature.

Place C: Change in the Release Mechanism

Changing the release mechanism mainly targets *total noise reduction*. In the SLR, we found the following approaches being used:

1. Test-and-release.
2. Sorting as an intermediary step.

Item 1: DSAT and DSFT uses thresholding to determine when to release data, as a way to save the privacy budget. Thresholding is particularly useful for dynamic data, as it often requires multiple releases over time. For example, adaptive or fixed thresholding can be used for sensor data and trajectory data, effectively providing a way of sampling the data.

SF also uses a type of test-and-release when creating the histogram structure using the exponential mechanism. The test-and-release approach means EM can be combined with basically any other release mechanism, which is also what we found in the literature. We believe the main challenge with EM is finding an adequate scoring/utility function, and this is where we believe a lot of accuracy improvement will come from.

Item 2 SORTaki is designed to be composable with two-step algorithms that release histograms, for example NF. The idea is that by sorting noisy values, they can group together similar values that would otherwise not be grouped due to the bins not being adjacent.

Place D: Pre-processing

Pre-processing generally targets *dimensionality reduction* or *total noise reduction*. In our SLR, we encountered the following types of pre-processing:

1. Encoding through projection/transformation.
2. Learning on non-sensitive data.

Item 1: Several algorithms project or transform their data, for example Privelet and EFPA. Encoding can reduce both sensitivity and dimensionality by decreasing redundancy, and is therefore especially interesting for multi-dimensional as well as high-dimensional, sparse, data sets. However, lossy compression

techniques can potentially introduce new sources of noise, and therefore adds another trade-off that needs to be taken into account. Intuitively, lossy compression is beneficial when the noise lost in the compression step is greater than the proportion of useful data points lost. For example, sparse data may benefit more from lossy compression than data that is not sparse.

Item 2: Delphi is a pre-processing step which uses a non-sensitive, public data set to build a decision tree. In cases where public data sets are available, it could be possible to adopt the same idea; for example learning a histogram structure on public data as opposed to spending budget on it. The caveat here is of course that the public data needs to be similar enough to the data used in the differentially private analysis, because otherwise this becomes an added source of noise. Thus, learning from non-sensitive data introduces another trade-off that is still largely unexplored.

7.6.2 Incomparable papers

We present a list of papers that were excluded during our qualitative analysis, and the reason for why we decided to exclude them in Section 7.5. The reason for excluding papers in the analysis step is that certain properties of their algorithms make them incomparable with other algorithms.

[106]: The DP-FC algorithm does not consider the structure of a histogram a sensitive attribute, and thus achieves a trivial accuracy improvement over other algorithms.

[107]: The APG algorithm does not perform differentially private clustering, and therefore achieves better accuracy by relaxing the privacy guarantees compared to AHP, IHP and GS.

[108]: The SC algorithm uses the ordering of the bins in order to calculate the cluster centers, but does not perturb the values before doing so, and thus the order is not protected, making their guarantees incomparable.

[109]: The Outlier-Histopub algorithm, similarly sorts the bin counts accord-

ing to size, without using the privacy budget accordingly to learn this information. The authors claim that this type of sorting does not violate differential privacy, but due to the fact that the output is determined based on the private data, the approach cannot be 0-differentially private.

[110]: The ASDP-HPA algorithm does not describe the details of how their use of Autoregressive Integrated Moving Average Model (ARIMA) is made private, and thus we cannot determine whether the entire algorithm is differentially private. Furthermore, the details of how they pre-process their data set is not divulged, and it can thus not be determined if the pre-processing violates differential privacy or not by changing the query sensitivity.

[111]: The algorithm is incomplete, since it only covers the histogram partitioning, and does not involve the addition of noise to bins. Furthermore, it is not clear whether they draw noise twice using the same budget, or if they reuse the same noise for their thresholds. As the privacy guarantee ε cannot be unambiguously deduced, we do not include their paper in our comparison.

[112]: The GBLUE algorithm generates a k-range tree based on the private data, where k is the fanout of the tree. Since private data is used to decide on whether a node is further split or not, it does not provide the same privacy guarantees as the other studied algorithms.

[113]: The algorithm creates groups based on the condition that the merged bins guarantee k-indistinguishability. Since this merge condition is based on the property of the data it does not guarantee differential privacy on the same level as the other papers, so we deem it incomparable.

Further, in the analysis regarding dimensions of accuracy improvement techniques presented in Section 7.5, some algorithms such as ADMM, SORTaki and Pythia are excluded. The rationale behind the exclusion is, these algorithms are not self contained, but nevertheless improves accuracy of the differentially private answers when combined with other analyzed algorithms.

Efforts such as Pythia and DPBench [114], that provide practitioners a way to em-

pirically assess the privacy/accuracy trade-off related to their data sets are commendable. However, to effectively use the tool one needs to have some background knowledge of the right combination of parameters to tune. In our analysis of the algorithms, we mapped out the accuracy improvement techniques grouped by optimization goals and corresponding query size. This knowledge will allow practitioners and researchers alike to think about other places to explore for accuracy improvement, rather than finding the algorithms that are based only on their data. Essentially, we provide an understanding to enable algorithm design, as opposed to algorithm selection.

7.7 Conclusions

Motivated by scarcity of works that structure knowledge concerning accuracy improvement in differentially private computations, we conducted a systematic literature review (SLR) on accuracy improvement techniques for histogram and synthetic data publication under differential privacy.

We present two results from our analysis that addresses our research objective, namely to synthesize the understanding of the underlying foundations of the privacy/accuracy trade-off in differentially private computations. This systematization of knowledge (SoK) includes:

1. Internal/external positioning of the studied algorithms (Figure 7.3 and Table 7.7).
2. A taxonomy of different *categories* (Figure 7.4) and their corresponding *optimization goals* to achieve accuracy improvement: *total noise reduction* (Table 7.8), *sensitivity reduction* (Table 7.9) and *data dimensionality reduction* (Table 7.10).

What's more, we also discuss and present an overview of composable algorithms according to their optimization goals and category, sort-out by the *places*, in which they operate (Section 7.6.1). Our intent is that these findings will pave

the way for future research by allowing others to integrate new solutions according to the categories. For example, our places can be used to reason about where to plug in new or existing techniques targeting a desired *optimization goal* during algorithm design.

From our overview of composability, we see that most efforts are focused on making *changes in the mechanism*, and on *post-processing*. We observe that, *altering the query* in one way or another, is not popular, and we believe further investigation is required to understand which techniques can be adopted in this place.

Finally, although all algorithms focus on accuracy improvement, it is impossible to select the 'best' algorithm without context. Intuitively, newer algorithms will have improved some property of an older algorithm, meaning that newer algorithms *may* provide higher accuracy. Still, the algorithms are used for different analyses, which means not all algorithms will be interchangeable. Secondly, many algorithms are data dependent, which means that the selection of the 'best' algorithm may change depending on the input data used, even when the analysis is fixed. Consequently, the 'best' algorithm needs to be chosen with a given data set and a given analysis in mind. The problem of choosing the 'best' algorithm when the setting is known is in fact addressed by Pythia.

7.7 Acknowledgements

Boel Nelson was partly funded by the Swedish Foundation for Strategic Research (SSF) and the Swedish Research Council (VR).

Bibliography

[1] P. Samarati and L. Sweeney. *Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression*. Tech. rep. SRI International, 1998.

[2] A. Machanavajjhala et al. “L-Diversity: Privacy beyond k -Anonymity”. English. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1.1 (2007).

[3] N. Li et al. “T-Closeness: Privacy Beyond k-Anonymity and l-Diversity”. In: *ICDE ’14*. 2007.

[4] A. Narayanan and V. Shmatikov. “Myths and Fallacies of "Personally Identifiable Information"”. In: *Commun. ACM* 53.6 (June 2010), pp. 24–26. (Visited on 01/27/2017).

[5] C. Dwork. “Differential Privacy”. en. In: *Automata, Languages and Programming*. Ed. by M. Bugliesi et al. Lecture Notes in Computer Science 4052. Springer Berlin Heidelberg, Jan. 2006, pp. 1–12.

[6] C. Dwork et al. “Calibrating Noise to Sensitivity in Private Data Analysis”. In: *Theory of Cryptography*. Ed. by S. Halevi and T. Rabin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 265–284.

[7] C. M. Bowen and F. Liu. “Comparative Study of Differentially Private Data Synthesis Methods”. arXiv preprint arXiv:1911.12704, 2016.

[8] H. Li et al. “Differentially Private Histogram and Synthetic Data Publication”. In: *Medical Data Privacy Handbook*. Ed. by A. Gkoulalas-

Divanis and G. Loukides. Cham: Springer, 2015, pp. 35–58. URL: https://doi.org/10.1007/978-3-319-23633-9_3 (visited on 10/15/2019).

[9] X. Meng et al. “Different Strategies for Differentially Private Histogram Publication”. In: *Journal of Communications and Information Networks* 2.3 (Sept. 1, 2017), pp. 68–77. URL: https://doi.org/10.1007/s41650-017-0014-x (visited on 10/15/2019).

[10] C. Dwork et al. “Our Data, Ourselves: Privacy Via Distributed Noise Generation”. In: *International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*. Ed. by S. Vaudenay. Springer, 2006.

[11] S. Meiser. “Approximate and Probabilistic Differential Privacy Definitions”. In: *IACR Cryptology ePrint Archive* (2018), p. 9.

[12] C. Dwork and A. Roth. “The Algorithmic Foundations of Differential Privacy”. In: *Foundations and Trends in Theoretical Computer Science* 9.3-4 (2014), pp. 211–407. URL: http://www.nowpublishers.com/articles/foundations-and-trends-in-theoretical-computer-science/TCS-042 (visited on 06/14/2016).

[13] F. McSherry and K. Talwar. “Mechanism Design via Differential Privacy”. In: *48th Annual IEEE Symposium on Foundations of Computer Science, 2007. FOCS ’07*. 48th Annual IEEE Symposium on Foundations of Computer Science, 2007. FOCS ’07. Oct. 2007, pp. 94–103.

[14] C. Dwork. “Differential Privacy: A Survey of Results”. In: *Theory and Applications of Models of Computation*. Ed. by M. Agrawal et al. Lecture Notes in Computer Science 4978. Springer Berlin Heidelberg, Jan. 1, 2008, pp. 1–19.

[15] M. Hay et al. “Accurate Estimation of the Degree Distribution of Private Networks”. In: *International Conference on Data Mining (ICDM)*. IEEE, 2009.

[16] C. Dwork et al. “Pan-Private Streaming Algorithms”. In: *ICS*. 2010, pp. 66–80.

[17] C. Dwork et al. “Differential privacy under continual observation”. In: Proceedings of the forty-second ACM symposium on Theory of computing. ACM, 2010, pp. 715–724. (Visited on 07/12/2016).

[18] G. Kellaris et al. “Differentially Private Event Sequences over Infinite Streams”. In: *Proc. VLDB Endow.* 7.12 (2014), pp. 1155–1166.

[19] B. Kitchenham. *Procedures for performing systematic reviews*. Joint Technical Report. Keele, UK: Software Engineering Group Department of Computer Science Keele University, UK, and Empirical Software Engineering, National ICT Australia Ltd, 2004, p. 26.

[20] B. A. Kitchenham et al. “Evidence-Based Software Engineering”. In: *Proceedings. 26th International Conference on Software Engineering*. IEEE, 2004, pp. 273–281.

[21] A. Sinha et al. “An Overview of Microsoft Academic Service (MAS) and Applications”. In: *International Conference on World Wide Web (WWW)*. ACM, 2015.

[22] Microsoft. *Microsoft Academic*. 2019. URL: `https://academic.microsoft.com/home` (visited on 10/09/2019).

[23] A.-W. Harzing. “Microsoft Academic (Search): A Phoenix Arisen from the Ashes?” en. In: *Scientometrics* 108.3 (Sept. 2016), pp. 1637–1647.

[24] A.-W. Harzing and S. Alakangas. “Microsoft Academic Is One Year Old: The Phoenix Is Ready to Leave the Nest”. en. In: *Scientometrics* 112.3 (Sept. 2017), pp. 1887–1894.

[25] A.-W. Harzing and S. Alakangas. “Microsoft Academic: Is the Phoenix Getting Wings?” en. In: *Scientometrics* 110.1 (Jan. 2017), pp. 371–383.

[26] S. E. Hug and M. P. Brändle. “The Coverage of Microsoft Academic: Analyzing the Publication Output of a University”. en. In: *Scientometrics* 113.3 (Dec. 2017), pp. 1551–1571.

[27] A.-W. Harzing. “Two New Kids on the Block: How Do Crossref and Dimensions Compare with Google Scholar, Microsoft Academic, Scopus and the Web of Science?” en. In: *Scientometrics* 120.1 (July 2019), pp. 341–349.

[28] R. C. Nickerson et al. “A Method for Taxonomy Development and Its Application in Information Systems”. In: *European Journal of Information Systems* 22.3 (May 1, 2013), pp. 336–359. URL: https://doi.org/10.1057/ejis.2012.26 (visited on 03/03/2020).

[29] M. Hay et al. “Boosting the Accuracy of Differentially Private Histograms through Consistency”. In: *International Conference on Very Large Data Bases (VLDB)*. ACM, 2010.

[30] B. Ding et al. “Differentially Private Data Cubes: Optimizing Noise Sources and Consistency”. In: *International Conference on Management of Data (SIGMOD)*. ACM, 2011.

[31] X. Xiao et al. “Differential Privacy via Wavelet Transforms”. In: *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 23.8 (2011), pp. 1200–1214.

[32] G. Acs et al. “Differentially Private Histogram Publishing through Lossy Compression”. In: *2012 IEEE 12th International Conference on Data Mining (ICDM)*. 2012 IEEE 12th International Conference on Data Mining (ICDM). 2012, pp. 1–10.

[33] J. Xu et al. “Differentially Private Histogram Publication”. In: *The VLDB Journal* 22.6 (Dec. 2013), pp. 797–822. URL: http://dx.doi.org/10.1007/s00778-013-0309-y (visited on 10/30/2015).

[34] H. Li et al. “Differentially Private Synthesization of Multi-Dimensional Data Using Copula Functions”. In: *International Conference on Extending Database Technology (EDBT)*. Vol. 2014. NIH Public Access, 2014.

[35] W. Lu et al. “Generating private synthetic databases for untrusted system evaluation”. In: *2014 IEEE 30th International Conference on Data Engineering (ICDE)*. 2014 IEEE 30th International Conference on Data Engineering (ICDE). 2014, pp. 652–663.

[36] Y. Park and J. Ghosh. “PeGS: Perturbed Gibbs Samplers That Generate Privacy-Compliant Synthetic Data”. In: *Transactions on Data Privacy (TDP)* 7.3 (2014), pp. 253–282.

[37] Y. Xiao et al. “DPCube: Differentially Private Histogram Release through Multidimensional Partitioning”. In: *Transactions on Data Privacy (TDP)* 7.3 (2014), pp. 195–222.

[38] J. Zhang et al. “PrivBayes: Private Data Release via Bayesian Networks”. In: *International Conference on Management of Data (SIGMOD)*. ACM, 2014.

[39] X. Zhang et al. “Towards Accurate Histogram Publication under Differential Privacy”. In: *International Conference on Data Mining (SDM)*. SIAM, 2014.

[40] R. Chen et al. “Private Analysis of Infinite Data Streams via Retroactive Grouping”. In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. CIKM ’15. Melbourne, Australia: Association for Computing Machinery, 2015, pp. 1061–1070. URL: https://doi.org/10.1145/2806416.2806454.

[41] J. Lee et al. “Maximum Likelihood Postprocessing for Differential Privacy under Consistency Constraints”. In: *International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2015.

[42] H. Li et al. “Differentially Private Histogram Publication for Dynamic Datasets: An Adaptive Sampling Approach”. In: *International on Conference on Information and Knowledge Management (CIKM)*. ACM, 2015.

[43] W.-Y. Day et al. “Publishing Graph Degree Distribution with Node Differential Privacy”. In: *International Conference on Management of Data (SIGMOD)*. ACM, 2016.

[44] S. Wang et al. “Private Weighted Histogram Aggregation in Crowdsourcing”. In: *International Conference on Wireless Algorithms, Systems, and Applications (WASA)*. Springer, 2016.

[45] J. Zhang et al. “PrivTree: A Differentially Private Algorithm for Hierarchical Decompositions”. In: *International Conference on Management of Data (SIGMOD)*. ACM, 2016.

[46] T. Benkhelif et al. “Co-Clustering for Differentially Private Synthetic Data Generation”. In: *International Workshop on Personal Analytics and Privacy (PAP)*. Springer, 2017.

[47] S. Doudalis and S. Mehrotra. “SORTaki: A Framework to Integrate Sorting with Differential Private Histogramming Algorithms”. In: *Conference on Privacy, Security and Trust (PST)*. IEEE, 2017.

[48] I. Kotsogiannis et al. “Pythia: Data Dependent Differentially Private Algorithm Selection”. In: *SIGMOD’17*. 2017.

[49] N. Wang et al. “Differentially Private Event Histogram Publication on Sequences over Graphs”. In: *Journal of Computer Science and Technology* 32.5 (2017), pp. 1008–1024.

[50] C. Xu et al. “DPPro: Differentially Private High-Dimensional Data Release via Random Projection”. In: *IEEE Transactions on Information Forensics and Security* 12.12 (2017), pp. 3081–3093.

[51] X. Ding et al. “Privacy-Preserving Triangle Counting in Large Graphs”. In: *International Conference on Information and Knowledge Management (CIKM)*. ACM, 2018.

[52] R. Gao and X. Ma. “Dynamic Data Histogram Publishing Based on Differential Privacy”. In: *2018 IEEE Intl Conf on Parallel Distributed Processing with Applications, Ubiquitous Computing Communications, Big Data Cloud Computing, Social Computing Networking, Sustainable Computing Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*. Melbourne, VIC, Australia: IEEE, 2018, pp. 737–743.

[53] S. Ghane et al. “Publishing Spatial Histograms under Differential Privacy”. In: *International Conference on Scientific and Statistical Database Management (SSDBM)*. ACM, 2018.

[54] H. Li et al. “IHP: Improving the Utility in Differential Private Histogram Publication”. In: *Distributed and Parallel Databases* 37 (2019), pp. 721–750.

[55] Y. Nie et al. “A Utility-Optimized Framework for Personalized Private Histogram Estimation”. In: *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 31.4 (Apr. 2019), pp. 655–669.

[56] X. Xiao et al. “Differential Privacy via Wavelet Transforms”. In: *International Conference on Data Engineering (ICDE)*. IEEE, 2010.

[57] J. Xu et al. “Differentially Private Histogram Publication”. In: *International Conference on Data Engineering (ICDE)*. IEEE, 2012.

[58] Y. Park et al. “Perturbed Gibbs Samplers for Generating Large-Scale Privacy-Safe Synthetic Health Data”. In: *International Conference on Healthcare Informatics*. IEEE, Sept. 2013.

[59] Y. Xiao et al. “Differentially Private Data Release through Multidimensional Partitioning”. In: *Workshop on Secure Data Management (SDM)*. Springer, 2010.

[60] Y. Xiao et al. “DPCube: Releasing Differentially Private Data Cubes for Health Information”. In: *International Conference on Data Engineering (ICDE)*. Arlington, VA, USA: IEEE, Apr. 2012, pp. 1305–1308.

[61] H. Li et al. “Improving the Utility in Differential Private Histogram Publishing: Theoretical Study and Practice”. In: *International Conference on Big Data (IEEE Big Data)*. IEEE, 2016.

[62] S. Boyd et al. “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers”. In: *Foundations and Trends® in Machine Learning* 3.1 (July 2011), pp. 1–122.

[63] A. Arasu et al. “Data Generation Using Declarative Constraints”. In: *International Conference on Management of Data (SIGMOD)*. ACM, 2011. URL: http://doi.acm.org/10.1145/1989323.1989395 (visited on 10/09/2019).

[64] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[65] A. Blum et al. “A Learning Theory Approach to Non-Interactive Database Privacy”. In: *Symposium on Theory of Computing (STOC)*. ACM, 2008.

[66] R. Chen et al. “Differentially Private Transit Data Publication: A Case Study on the Montreal Transportation System”. In: *International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2012.

URL: http://doi.acm.org/10.1145/2339530.2339564 (visited on 10/09/2019).

[67] X. Xiao et al. "Optimal Random Perturbation at Multiple Privacy Levels". In: *International Conference on Very Large Data Bases (VLDB)*. ACM, Aug. 1, 2009. URL: http://dl.acm.org/citation.cfm?doid=1687627.1687719 (visited on 10/09/2019).

[68] A. Machanavajjhala et al. "Privacy: Theory Meets Practice on the Map". In: *International Conference on Data Engineering (ICDE)*. IEEE, 2008. URL: https://doi.org/10.1109/ICDE.2008.4497436 (visited on 07/16/2018).

[69] H. Xie et al. "Distributed Histograms for Processing Aggregate Data from Moving Objects". In: *International Conference on Mobile Data Management (MDM)*. IEEE, 2007.

[70] J. C. Duchi et al. "Local Privacy and Statistical Minimax Rates". In: *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. 2013 IEEE 54th Annual Symposium on Foundations of Computer Science. Oct. 2013, pp. 429–438.

[71] J. Blocki et al. "Differentially Private Data Analysis of Social Networks via Restricted Sensitivity". In: *Conference on Innovations in Theoretical Computer Science (ITCS)*. ACM, 2013. URL: http://doi.acm.org/10.1145/2422436.2422449 (visited on 10/09/2019).

[72] C. Li and G. Miklau. "An Adaptive Mechanism for Accurate Query Answering Under Differential Privacy". In: *International Conference on Very Large Data Bases (VLDB)*. ACM, Feb. 2012. URL: https://doi.org/10.14778/2168651.2168653 (visited on 10/09/2019).

[73] S. Raskhodnikova and A. Smith. "Efficient Lipschitz Extensions for High-Dimensional Graph Statistics and Node Private Degree Distributions". arXiv preprint arXiv:1504.07912, Apr. 29, 2015. arXiv: 1504.07912 [cs]. URL: http://arxiv.org/abs/1504.07912 (visited on 10/09/2019).

[74] B. Barak et al. “Privacy, Accuracy, and Consistency Too: A Holistic Solution to Contingency Table Release”. In: *Symposium on Principles of Database Systems (PODS)*. ACM, 2007.

[75] C. Zeng et al. “On Differentially Private Frequent Itemset Mining”. In: *International Conference on Very Large Data Bases (VLDB)*. ACM, Jan. 2012.

[76] A. Ghosh et al. “Universally Utility-Maximizing Privacy Mechanisms”. In: *SIAM Journal on Computing* 41.6 (Jan. 2012), pp. 1673–1693.

[77] G. Kellaris and S. Papadopoulos. “Practical Differential Privacy via Grouping and Smoothing”. In: *International Conference on Very Large Data Bases (VLDB)*. ACM, 2013. URL: http://dl.acm.org/citation.cfm?id=2488335.2488337 (visited on 11/23/2015).

[78] C. Li et al. “Optimizing Linear Counting Queries Under Differential Privacy”. In: *Symposium on Principles of Database Systems (PODS)*. ACM, 2010.

[79] M. Hardt et al. “A Simple and Practical Algorithm for Differentially Private Data Release”. In: *Advances in Neural Information Processing Systems (NIPS)*. Ed. by F. Pereira et al. Curran Associates, Inc., 2012. URL: http://papers.nips.cc/paper/4548-a-simple-and-practical-algorithm-for-differentially-private-data-release.pdf (visited on 10/09/2019).

[80] R. Chen et al. “Differentially Private Sequential Data Publication via Variable-Length N-Grams”. In: *Conference on Computer and Communications Security (CCS)*. ACM, 2012. URL: http://doi.acm.org/10.1145/2382196.2382263 (visited on 10/09/2019).

[81] M. Hardt and G. N. Rothblum. “A Multiplicative Weights Mechanism for Privacy-Preserving Data Analysis”. In: *Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2010.

[82] A. Inan et al. “Private Record Matching Using Differential Privacy”. In: *International Conference on Extending Database Technology (EDBT)*. ACM, 2010. URL: http://doi.acm.org/10.1145/1739041.1739059 (visited on 10/09/2019).

[83] G. Cormode et al. “Differentially Private Spatial Decompositions”. In: *2012 IEEE 28th International Conference on Data Engineering (ICDE)*. 2012 IEEE 28th International Conference on Data Engineering (ICDE). 2012, pp. 20–31.

[84] Ú. Erlingsson et al. “RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response”. In: *CCS’14*. 2014.

[85] V. Rastogi and S. Nath. “Differentially Private Aggregation of Distributed Time-Series with Transformation and Encryption”. In: *International Conference on Management of Data (SIGMOD)*. ACM, 2010. URL: http://doi.acm.org/10.1145/1807167.1807247 (visited on 11/06/2017).

[86] S. P. Kasiviswanathan et al. “Analyzing Graphs with Node Differential Privacy”. In: *Theory of Cryptography Conference (TCC)*. Ed. by A. Sahai. Springer, 2013. URL: http://link.springer.com/10.1007/978-3-642-36594-2_26 (visited on 10/09/2019).

[87] H. V. Jagadish et al. “Optimal histograms with quality guarantees”. In: *VLDB*. Vol. 98. 1998, pp. 24–27.

[88] G. Cormode et al. “Differentially Private Summaries for Sparse Data”. In: *International Conference on Database Theory (ICDT)*. ACM, 2012. URL: http://doi.acm.org/10.1145/2274576.2274608 (visited on 10/09/2019).

[89] A. Friedman and A. Schuster. “Data Mining with Differential Privacy”. In: *International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, July 2010.

[90] J. Zhang et al. “PrivGene: Differentially Private Model Fitting Using Genetic Algorithms”. In: *International Conference on Management of Data (SIGMOD)*. ACM, 2013.

[91] K. Chaudhuri et al. “Differentially Private Empirical Risk Minimization”. In: *Journal of Machine Learning Research* 12 (Mar 2011), pp. 1069–1109. URL: http://www.jmlr.org/papers/v12/chaudhuri11a.html (visited on 10/09/2019).

[92] L. Fan and L. Xiong. “An Adaptive Approach to Real-Time Aggregate Monitoring With Differential Privacy”. In: *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 26.9 (2014), pp. 2094–2106.

[93] G. Yuan et al. “Low-Rank Mechanism: Optimizing Batch Queries under Differential Privacy”. In: *International Conference on Very Large Data Bases (VLDB)*. ACM, July 2012.

[94] W. Qardaji et al. “Differentially private grids for geospatial data”. In: *2013 IEEE 29th International Conference on Data Engineering (ICDE)*. 2013 IEEE 29th International Conference on Data Engineering (ICDE). 2013, pp. 757–768.

[95] W. Qardaji et al. “Understanding Hierarchical Methods for Differentially Private Histograms”. In: *International Conference on Very Large Data Bases (VLDB)*. ACM, Sept. 2013. URL: http://dx.doi.org/10.14778/2556549.2556576 (visited on 10/09/2019).

[96] D. Su et al. “Differentially Private K-Means Clustering”. In: *Conference on Data and Application Security and Privacy (CODASPY)*. ACM, 2016. URL: http://doi.acm.org/10.1145/2857705.2857708 (visited on 10/09/2019).

[97] C. Li et al. “A Data- and Workload-Aware Algorithm for Range Queries Under Differential Privacy”. In: *International Conference on Very Large Data Bases (VLDB)*. ACM, Jan. 2014. URL: http://dx.doi.org/10.14778/2732269.2732271 (visited on 05/07/2019).

[98] X. He et al. “DPT: Differentially Private Trajectory Synthesis Using Hierarchical Reference Systems”. In: *International Conference on Very Large Data Bases (VLDB)*. ACM, Jan. 2015.

[99] B. I. P. Rubinstein et al. “Learning in a Large Function Space: Privacy-Preserving Mechanisms for SVM Learning”. In: *Journal of Privacy and Confidentiality* 4.1 (July 20, 2012). URL: http://www.journalprivacyconfidentiality.org/index.php/jpc/article/view/612 (visited on 10/09/2019).

[100] W. Qardaji et al. “PriView: Practical Differentially Private Release of Marginal Contingency Tables”. In: *International Conference on Management of Data (SIGMOD)*. ACM, 2014. URL: http://doi.acm.org/10.1145/2588555.2588575 (visited on 10/09/2019).

[101] R. Chen et al. “Differentially Private High-Dimensional Data Publication via Sampling-Based Inference”. In: *International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2015. URL: http://doi.acm.org/10.1145/2783258.2783379 (visited on 10/09/2019).

[102] R. Bassily and A. Smith. “Local, Private, Efficient Protocols for Succinct Histograms”. In: *Symposium on Theory of Computing (STOC)*. ACM, 2015. URL: http://doi.acm.org/10.1145/2746539.2746632 (visited on 04/11/2017).

[103] M. Boullé. “Data Grid Models for Preparation and Modeling in Supervised Learning”. In: *Hands-On Pattern Recognition: Challenges in Machine Learning* 1 (2011), pp. 99–130.

[104] P. J. Rousseeuw and G. Molenberghs. “Transformation of Non Positive Semidefinite Correlation Matrices”. In: *Communications in Statistics - Theory and Methods* 22.4 (1993), pp. 965–984. URL: https://doi.org/10.1080/03610928308831068.

[105] K. Weinberger et al. “Feature Hashing for Large Scale Multitask Learning”. In: *International Conference on Machine Learning (ICML)*. The 26th Annual International Conference. ACM, 2009. URL: http://portal.acm.org/citation.cfm?doid=1553374.1553516 (visited on 02/28/2020).

[106] F. Yan et al. “Differentially Private Histogram Publishing through Fractal Dimension for Dynamic Datasets”. In: *Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, 2018.

[107] X. Liu and S. Li. “Histogram Publishing Method Based on Differential Privacy”. In: *International Conference on Computer Science and Software Engineering (CSSE)* (2018).

[108] Q. Qian et al. “Publishing Graph Node Strength Histogram with Edge Differential Privacy”. In: *International Conference on Database Systems for Advanced Applications (DASFAA)*. Springer, 2018.

[109] Q. Han et al. “Publishing Histograms with Outliers under Data Differential Privacy”. In: *Security and Communication Networks* 9.14 (2016), pp. 2313–2322.

[110] Y. Li and S. Li. “Research on Differential Private Streaming Histogram Publication Algorithm”. In: *International Conference on Cloud Computing and Intelligence Systems (CCIS)*. IEEE, 2018.

[111] M. Hadian et al. “Privacy-Preserving mHealth Data Release with Pattern Consistency”. In: *Global Communications Conference (GLOBECOM)*. IEEE, 2016.

[112] H. Chen et al. “An Iterative Algorithm for Differentially Private Histogram Publication”. In: *International Conference on Cloud Computing and Big Data (CLOUDCOM-ASIA)*. IEEE, 2013.

[113] X. Li et al. “Differential Privacy for Edge Weights in Social Networks”. In: *Security and Communication Networks* 2017 (2017), pp. 1–10.

[114] M. Hay et al. “Principled Evaluation of Differentially Private Algorithms Using DPBench”. In: *Proceedings of the 2016 International Conference on Management of Data*. SIGMOD ’16. New York, NY, USA: ACM, 2016, pp. 139–154.

[115] V. Balcer and S. P. Vadhan. “Differential Privacy on Finite Computers”. In: *Conference on Innovations in Theoretical Computer Science (ITCS)*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018.

[116] T. Benkhelif. “Publication de Données Individuelles Respectueuse de La Vie Privée : Une Démarche Fondée Sur Le Co-Clustering”. PhD thesis. Université de Nantes, 2018.

[117] A. Bhowmick et al. “Differential Privacy Using a Multibit Histogram”. U.S. pat. 20180349620A1. Apple Inc. Dec. 6, 2018. URL: https://patents.google.com/patent/US20180349620A1/en?q=DIFFERENTIAL&q=PRIVACY&q=USING&q=A&q=MULTIBIT&

q=HISTOGRAM&oq=DIFFERENTIAL+PRIVACY+USING+A+MULTIBIT+HISTOGRAM (visited on 02/11/2020).

[118] C. M. Bowen and F. Liu. "Differentially Private Release and Analysis of Youth Voter Registration Data via Statistical Election to Partition Sequentially". arXiv preprint arXiv:1602.01063, Mar. 18, 2018. arXiv: 1803.06763 [stat]. URL: http://arxiv.org/abs/1803.06763 (visited on 06/27/2019).

[119] C. M. Bowen and F. Liu. "STatistical Election to Partition Sequentially (STEPS) and Its Application in Differentially Private Release and Analysis of Youth Voter Registration Data". arXiv preprint arXiv:1803.06763, 2018.

[120] K. Chaudhuri and S. A. Vinterbo. "A Stability-Based Validation Procedure for Differentially Private Machine Learning". In: *Advances in Neural Information Processing Systems (NIPS)*. Curran Associates, Inc., 2013.

[121] B. Cyphers and K. Veeramachaneni. "AnonML: Locally Private Machine Learning over a Network of Peers". In: *International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2017.

[122] E. C. Eugenio and F. Liu. "CIPHER: Construction of dIfferentially Private Microdata from Low-Dimensional Histograms via Solving Linear Equations with Tikhonov Regularization." arXiv preprint arXiv:1812.05671, 2018.

[123] M. Fanaeepour et al. "The CASE Histogram: Privacy-Aware Processing of Trajectory Data Using Aggregates". In: *Geoinformatica* 19.4 (2015), pp. 747–798.

[124] M. Fanaeepour and B. I. P. Rubinstein. "End-to-End Differentially-Private Parameter Tuning in Spatial Histograms." arXiv preprint arXiv:1702.05607, 2017.

[125] M. Fanaeepour and B. I. P. Rubinstein. "Histogramming Privately Ever After: Differentially-Private Data-Dependent Error Bound Optimisation". In: *International Conference on Data Engineering (ICDE)*. IEEE, 2018.

[126] Y. Fei et al. “Differential Privacy Protection-Based Data Release Method for Spark Framework”. Pat. CN107766740A (China). 2018.

[127] A. Foote et al. *Releasing Earnings Distributions Using Differential Privacy: Disclosure Avoidance System For Post Secondary Employment Outcomes (PSEO)*. Economic Analysis. 2019.

[128] J. J. Gardner et al. “SHARE: System Design and Case Studies for Statistical Health Information Release”. In: *Journal of the American Medical Informatics Association* 20.1 (2013), pp. 109–116.

[129] J. Gehrke et al. “Crowd-Blending Privacy”. In: *International Cryptology Conference (CRYPTO)*. Ed. by R. Safavi-Naini and R. Canetti. Springer, 2012.

[130] R. Hall et al. “Random Differential Privacy”. arXiv preprint arXiv:1112.2680, 2013.

[131] M. Hardt and K. Talwar. “On the Geometry of Differential Privacy”. In: *Symposium on Theory of Computing (STOC)*. ACM, 2010.

[132] G. Kellaris et al. “Engineering Methods for Differentially Private Histograms: Efficiency Beyond Utility”. In: *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 31.2 (2019), pp. 315–328.

[133] Y. N. Kobliner et al. “Locally Private Determination of Heavy Hitters”. U.S. pat. 20180336357A1. Harvard College, University of California, Georgetown University. Nov. 22, 2018. URL: https://patents.google.com/patent/US20180336357A1/en?q=LOCALLY&q=PRIVATE&q=DETERMINATION&q=OF&q=HEAVY&q=HITTERS&oq=LOCALLY+PRIVATE+DETERMINATION+OF+HEAVY+HITTERS (visited on 02/11/2020).

[134] T. Kulkarni et al. “Answering Range Queries Under Local Differential Privacy”. arXiv preprint arXiv:1812.10942, 2018.

[135] S. Lan et al. “Greedy Algorithm Based on Bucket Partitioning for Differentially Private Histogram Publication”. In: *Journal of Xiamen University* (2013).

[136] J. Lei. “Differentially Private M-Estimators”. In: *Advances in Neural Information Processing Systems (NIPS)*. Curran Associates, Inc., 2011.

[137] H. Li et al. “Efficient E-Health Data Release with Consistency Guarantee under Differential Privacy”. In: *International Conference on E-Health Networking, Application & Services (HealthCom)*. IEEE, 2015.

[138] B. Li et al. “A Privacy Preserving Algorithm to Release Sparse High-Dimensional Histograms”. In: *Journal of Privacy and Confidentiality* 8.1 (2018).

[139] X. Li et al. “Differentially Private Release of the Distribution of Clustering Coefficients across Communities”. In: *Security and Communication Networks* 2019 (2019), pp. 1–9.

[140] Y. Li et al. “Impact of Prior Knowledge and Data Correlation on Privacy Leakage: A Unified Analysis”. In: *IEEE Transactions on Information Forensics and Security* 14.9 (2019), pp. 2342–2357.

[141] B.-R. Lin and D. Kifer. “Information Preservation in Statistical Privacy and Bayesian Estimation of Unattributed Histograms”. In: *International Conference on Management of Data (SIGMOD)*. ACM, 2013.

[142] G. Ling et al. “Detrended Analysis Differential Privacy Protection-Based Histogram Data Release Method”. Pat. CN108446568A (China). 2018.

[143] C. Luo et al. “Predictable Privacy-Preserving Mobile Crowd Sensing: A Tale of Two Roles”. In: *IEEE/ACM Transactions on Networking* 27.1 (2019), pp. 361–374.

[144] E. Naghizade et al. “Challenges of Differentially Private Release of Data Under an Open-World Assumption”. In: *International Conference on Scientific and Statistical Database Management (SSDBM)*. ACM, 2017.

[145] A. Nikolov et al. “The Geometry of Differential Privacy: The Small Database and Approximate Cases”. In: *SIAM Journal on Computing* 45.2 (2016), pp. 575–616.

[146] J. Raigoza. “Differential Private-Hilbert: Data Publication Using Hilbert Curve Spatial Mapping”. In: *International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2017.

[147] A. Roth. “Differential Privacy and the Fat-Shattering Dimension of Linear Queries”. In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, 2010, pp. 683–695.

[148] S. Shang et al. “The Application of Differential Privacy for Rank Aggregation: Privacy and Accuracy”. In: *International Conference on Information Fusion (FUSION)*. IEEE, 2014.

[149] D. B. Smith et al. “More Flexible Differential Privacy: The Application of Piecewise Mixture Distributions in Query Release”. arXiv preprint arXiv:1707.01189, 2017.

[150] D. Su et al. “PrivPfC: Differentially Private Data Publication for Classification”. In: *International Conference on Very Large Data Bases (VLDB)*. ACM, 2018.

[151] X. Xiao et al. “iReduct: Differential Privacy with Reduced Relative Errors”. In: *International Conference on Management of Data (SIGMOD)*. ACM, 2011.

[152] X. Xiaoling et al. “Histogram-Based Data Flow-Oriented Differential Privacy Publishing Method”. Pat. CN105046160A (China). 2015.

[153] X. Ying et al. “On Linear Refinement of Differential Privacy-Preserving Query Answering”. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*. Springer, 2013.

[154] L. Zhang et al. “Differentially Private Linear Queries on Histograms”. U.S. pat. 9672364B2. Microsoft Technology Licensing LLC. June 6, 2017. URL: https://patents.google.com/patent/US9672364B2/en?q=Differentially&q=private&q=linear&q=queries&q=histograms&oq=Differentially+private+linear+queries+on+histograms (visited on 02/14/2020).

[155] T. Zhu et al. “Correlated Differential Privacy: Hiding Information in Non-IID Data Set”. In: *IEEE Transactions on Information Forensics and Security* 10.2 (2015), pp. 229–242.

[156] N. C. Abay et al. “Privacy Preserving Synthetic Data Release Using Deep Learning”. In: *Joint European Conference on Machine Learn-*

ing and Knowledge Discovery in Databases (ECML PKDD). Springer, 2018.

[157] J. M. Abowd and L. Vilhuber. "How Protective Are Synthetic Data". In: *International Conference on Privacy in Statistical Databases (PSD)*. Springer, 2008.

[158] M. Aliakbarpour et al. "Differentially Private Identity and Closeness Testing of Discrete Distributions." arXiv preprint arXiv:1707.05497, 2017.

[159] M. Balog et al. "Differentially Private Database Release via Kernel Mean Embeddings". In: *International Conference on Machine Learning (ICML)*. ACM, 2018.

[160] A. F. Barrientos et al. "Providing Access to Confidential Research Data through Synthesis and Verification: An Application to Data on Employees of the U.S. Federal Government". In: *The Annals of Applied Statistics (AOAS)* 12.2 (2018), pp. 1124–1156.

[161] A. F. Barrientos et al. "Differentially Private Significance Tests for Regression Coefficients". In: *Journal of Computational and Graphical Statistics* 28.2 (2018), pp. 1–24.

[162] V. Bindschaedler et al. "Plausible Deniability for Privacy-Preserving Data Synthesis". In: *International Conference on Very Large Data Bases (VLDB)*. ACM, 2017.

[163] A. Blum et al. "A Learning Theory Approach to Noninteractive Database Privacy". In: *Journal of the ACM* 60.2 (2013), p. 12.

[164] J. Böhler et al. "Privacy-Preserving Outlier Detection for Data Streams". In: *Conference on Data and Applications Security and Privacy (DBSec)*. IFIP WG 11.3, 2017.

[165] O. Bousquet et al. "Passing Tests without Memorizing: Two Models for Fooling Discriminators". arXiv preprint arXiv:1902.03468, 2019.

[166] C. M. Bowen and F. Liu. "Differentially Private Data Synthesis Methods". arXiv preprint arXiv:1602.01063, 2016.

[167] Y. Cao et al. “Quantifying Differential Privacy under Temporal Correlations”. In: *International Conference on Data Engineering (ICDE)*. IEEE, 2017.

[168] Y. Cao et al. “PriSTE: From Location Privacy to Spatiotemporal Event Privacy.” arXiv preprint arXiv:1810.09152, 2018.

[169] A.-S. Charest. “How Can We Analyze Differentially-Private Synthetic Datasets?” In: *Journal of Privacy and Confidentiality* 2.2 (2011), p. 3.

[170] L. Chen et al. “WaveCluster with Differential Privacy”. In: *International Conference on Information and Knowledge Management (CIKM)*. ACM, 2015.

[171] G. Cormode et al. “Constrained Private Mechanisms for Count Data”. In: *International Conference on Data Engineering (ICDE)*. IEEE, 2018.

[172] C. Dwork et al. “On the Complexity of Differentially Private Data Release: Efficient Algorithms and Hardness Results”. In: *Symposium on Theory of Computing (STOC)*. ACM, 2009.

[173] M. J. Elliot. “Empirical Differential Privacy: An New Method for Measuring Residual Dis-Cloure Risk in Synthetic Data”. 2014.

[174] L. Fan and L. Xiong. “Adaptively Sharing Time-Series with Differential Privacy”. arXiv preprint arXiv:1202.3461, 2012.

[175] S. Garfinkel. *De-Identifying Government Datasets*. Technical Report. National Institute of Standards and Technology, 2016.

[176] S. Garfinkel. *De-Identifying Government Datasets (2nd Draft)*. Technical Report. National Institute of Standards and Technology, 2016.

[177] A. Gupta et al. “Iterative Constructions and Private Data Release”. In: *Theory of Cryptography Conference (TCC)*. Springer, 2012.

[178] M. A. W. Hardt. “A Study of Privacy and Fairness in Sensitive Data Analysis”. PhD thesis. Princeton University, 2011.

[179] Y. Hu et al. “Privacy-Preserving Task Allocation for Edge Computing Enhanced Mobile Crowdsensing”. In: *International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP)*. Springer, 2018.

[180] B. Hu et al. “PSCluster: Differentially Private Spatial Cluster Detection for Mobile Crowdsourcing Applications”. In: *Conference on Communications and Network Security (CNS)*. IEEE, 2018.

[181] J. Jordon et al. “PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees”. In: *International Conference on Learning Representations (ICLR)*. 2019.

[182] Z. Jorgensen et al. “Conservative or Liberal? Personalized Differential Privacy”. In: *International Conference on Data Engineering (ICDE)*. IEEE, 2015.

[183] D. Kifer and B. R. Lin. “Towards an Axiomatization of Statistical Privacy and Utility”. In: *Symposium on Principles of Database Systems (PODS)*. ACM, 2010.

[184] T. Kulkarni et al. “Constrained Differential Privacy for Count Data”. arXiv preprint arXiv:1710.00608, 2017.

[185] J. Lee. “On Sketch Based Anonymization That Satisfies Differential Privacy Model”. In: *Canadian Conference on Advances in Artificial Intelligence*. Springer, 2010.

[186] C. Li and G. Miklau. “Optimal Error of Query Sets under the Differentially-Private Matrix Mechanism”. In: *International Conference on Database Theory (ICDT)*. ACM, 2013.

[187] H. Li et al. “DPSynthesizer: Differentially Private Data Synthesizer for Privacy Preserving Data Sharing”. In: *International Conference on Very Large Data Bases (VLDB)*. ACM, 2014.

[188] C. Li and G. Miklau. “Lower Bounds on the Error of Query Sets Under the Differentially-Private Matrix Mechanism”. In: *Theory of Computing Systems* 57.4 (2015), pp. 1159–1201.

[189] M. Li and X. Ma. “Bayesian Networks-Based Data Publishing Method Using Smooth Sensitivity”. In: *International Conference on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing &*

Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom). IEEE, 2018.

[190] Y. Li et al. “Towards Differentially Private Truth Discovery for Crowd Sensing Systems”. arXiv preprint arXiv:1810.04760, 2018.

[191] F. Liu. “Model-Based Differentially Private Data Synthesis”. arXiv preprint arXiv:1606.08052, 2016.

[192] K.-C. Liu et al. “Optimized Data De-Identification Using Multidimensional k-Anonymity”. In: *International Conference On Trust, Security And Privacy In Computing and Communications/International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, 2018.

[193] P.-H. Lu and C.-M. Yu. “POSTER: A Unified Framework of Differentially Private Synthetic Data Release with Generative Adversarial Network”. In: *Conference on Computer and Communications Security (CCS)*. ACM, 2017.

[194] G. J. Matthews and O. Harel. “Data Confidentiality: A Review of Methods for Statistical Disclosure Limitation and Methods for Assessing Privacy”. In: *Statistics Surveys* 5 (2011), pp. 1–29.

[195] D. McClure and J. P. Reiter. “Differential Privacy and Statistical Disclosure Risk Measures: An Investigation with Binary Synthetic Data”. In: *Transactions on Data Privacy (TDP)* 5.3 (2012), pp. 535–552.

[196] D. R. McClure. “Relaxations of Differential Privacy and Risk/Utility Evaluations of Synthetic Data and Fidelity Measures”. PhD thesis. Duke University, 2015.

[197] Y. Mülle et al. “Privacy-Integrated Graph Clustering Through Differential Privacy.” In: *EDBT/ICDT Workshops*. 2015.

[198] S. Neel et al. “How to Use Heuristics for Differential Privacy”. arXiv preprint arXiv:1811.07765, 2018.

[199] M.-J. Park and H. J. Kim. “Statistical Disclosure Control for Public Microdata: Present and Future”. In: *Korean Journal of Applied Statistics* 29.6 (2016), pp. 1041–1059.

[200] H. Ping et al. “DataSynthesizer: Privacy-Preserving Synthetic Datasets”. In: *International Conference on Scientific and Statistical Database Management (SSDBM)*. ACM, 2017.

[201] L. Rodriguez and B. Howe. “Privacy-Preserving Synthetic Datasets Over Weakly Constrained Domains”. arXiv preprint arXiv:1808.07603, 2018.

[202] N. Shlomo. “Statistical Disclosure Limitation: New Directions and Challenges”. In: *Journal of Privacy and Confidentiality* 8.1 (2018).

[203] J. Snoke and A. B. Slavkovic. “pMSE Mechanism: Differentially Private Synthetic Data with Maximal Distributional Similarity”. In: *International Conference on Privacy in Statistical Databases (PSD)*. Springer, 2018.

[204] J. V. Snoke. “Statistical Data Privacy Methods for Increasing Research Opportunities”. PhD thesis. Pennsylvania State University, 2018.

[205] A. Triastcyn and B. Faltings. “Generating Differentially Private Datasets Using GANs”. arXiv preprint arXiv:1803.03148, 2018.

[206] J. R. Ullman. “Privacy and the Complexity of Simple Queries”. PhD thesis. Harvard, 2013.

[207] L. Vilhuber et al. “Synthetic Establishment Microdata around the World”. In: *Statistical Journal of the IAOS* 32.1 (2016), pp. 65–68.

[208] J. Wang et al. “Protecting Query Privacy with Differentially Private K-Anonymity in Location-Based Services”. In: *Personal and Ubiquitous Computing* 22.3 (2018), pp. 453–469.

[209] Z. Wang et al. “A Data Publishing System Based on Privacy Preservation”. In: *International Conference on Database Systems for Advanced Applications (DASFAA)*. Springer, 2019.

[210] B. Weggenmann and F. Kerschbaum. “SynTF: Synthetic and Differentially Private Term Frequency Vectors for Privacy-Preserving Text Mining”. In: *International Conference on Research and Development in Information Retrieval (SIGIR)*. ACM, 2018.

[211] C. Xu et al. “GANobfuscator: Mitigating Information Leakage under GAN via Differential Privacy”. In: *IEEE Transactions on Information Forensics and Security* 14.9 (2019), pp. 2358–2371.

[212] H. Yu. “Differentially Private Verification ofPredictions from Synthetic Data”. Master thesis. Duke University, 2017.

[213] J. Zhang. “Algorithms for Synthetic Data Release under Differential Privacy”. PhD thesis. Nanyang Technological University, 2016.

[214] X. Zhang et al. “Differentially Private Releasing via Deep Generative Model”. arXiv preprint arXiv:1801.01594, 2018.

[215] S. Zhou et al. “Differential Privacy with Compression”. In: *International Symposium on Information Theory (ISIT)*. IEEE, 2009.

7.A Excluded Papers

7.A.1 Query 1

Table 7.12: Excluded papers from query 1 (focusing on histograms), and the corresponding exclusion criteria.

Citation	Exclusion Criteria
Balcer and Vadhan [115]	2
Bassily and Smith [102]	6
Benkhelif [116]	9, 10
Bhowmick et al. [117]	7
Bowen and Liu [118]	8
Bowen and Liu [119]	8
Chaudhuri and Vinterbo [120]	5
Cyphers and Veeramachaneni [121]	5
Eugenio and Liu [122]	5
Fanaeepour et al. [123]	1

Continued on next page

Table 7.12 – *Continued from previous page*

Citation	**Exclusion Criteria**
Fanaeepour and Rubinstein [124]	4
Fanaeepour and Rubinstein [125]	2
Fei et al. [126]	7
Foote et al. [127]	5
Gardner et al. [128]	2
Gehrke et al. [129]	3
Hall et al. [130]	3
Hardt and Rothblum [81]	1, 2, 6
Hardt and Talwar [131]	6
Kellaris et al. [132]	2
Kobliner et al. [133]	7
Kulkarni et al. [134]	9
Lan et al. [135]	10
Lei [136]	5
Li et al. [78]	6
Li et al. [8]	2
Li et al. [137]	2
Li et al. [138]	2
Li et al. [139]	5
Li et al. [140]	1, 2, 5
Lin and Kifer [141]	1, 2, 5
Ling et al. [142]	7
Luo et al. [143]	1, 2, 3
Meng et al. [9]	2
Naghizade et al. [144]	1
Nikolov et al. [145]	6
Raigoza [146]	2, 6, 9
Roth [147]	6

Continued on next page

Table 7.12 – *Continued from previous page*

Citation	**Exclusion Criteria**
Shang et al. [148]	2
Smith et al. [149]	1
Su et al. [150]	5
Xiao et al. [151]	3
Xiaoling et al. [152]	7
Ying et al. [153]	2, 6
Zhang et al. [154]	7
Zhu et al. [155]	2

7.A.2 Query 2

Table 7.13: Excluded papers from query 2 (focusing on synthetic data), and the corresponding exclusion criteria.

Citation	Exclusion Criteria
Abay et al. [156]	2
Abowd and Vilhuber [157]	1
Aliakbarpour et al. [158]	1
Balog et al. [159]	2, 6
Barak et al. [74]	6
Barrientos et al. [160]	1
Barrientos et al. [161]	1
Bindschaedler et al. [162]	3
Blum et al. [65]	2
Blum et al. [163]	2
Böhler et al. [164]	4
Bousquet et al. [165]	1
Bowen and Liu [7]	2,6
Bowen and Liu [166]	8
Bowen and Liu [119]	8
Cao et al. [167]	1, 2
Cao et al. [168]	1
Charest [169]	1
Chen et al. [170]	1
Cormode et al. [171]	1, 2
Dwork et al. [172]	2, 6
Elliot [173]	8
Fan and Xiong [92]	1
Fan and Xiong [174]	8

Continued on next page

Table 7.13 – *Continued from previous page*

Citation	**Exclusion Criteria**
Garfinkel [175]	1
Garfinkel [176]	1
Gehrke et al. [129]	2
Gupta et al. [177]	2
Hardt [178]	9
Hu et al. [179]	1
Hu et al. [180]	1
Jordon et al. [181]	5
Jorgensen et al. [182]	1, 2
Kifer and Lin [183]	1
Kulkarni et al. [184]	8
Lee [185]	2
Li and Miklau [186]	1
Li et al. [187]	2
Li and Miklau [188]	1, 2
Li et al. [8]	2,6
Li and Ma [189]	4
Li et al. [138]	2
Li et al. [190]	4
Liu [191]	1
Liu et al. [192]	2
Lu and Yu [193]	2
Machanavajjhala et al. [68]	3, 6
Matthews and Harel [194]	2, 6
McClure and Reiter [195]	1
McClure [196]	9
Mülle et al. [197]	1
Neel et al. [198]	6

Continued on next page

Table 7.13 – *Continued from previous page*

Citation	**Exclusion Criteria**
Park and Kim [199]	10
Ping et al. [200]	2
Rodriguez and Howe [201]	2
Shlomo [202]	1
Snoke and Slavkovic [203]	2
Snoke [204]	9
Triastcyn and Faltings [205]	2
Ullman [206]	9
Vilhuber et al. [207]	1
Wang et al. [208]	1, 3, 5
Wang et al. [209]	2
Weggenmann and Kerschbaum [210]	1
Xu et al. [211]	1
Yu [212]	9
Zhang [213]	9
Zhang et al. [214]	5
Zhou et al. [215]	1