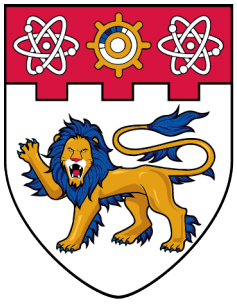


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

Privacy-preserving Data Analytics

ZHAO YANG

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

2022

PRIVACY-PRESERVING DATA ANALYTICS

ZHAO YANG

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2022

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

30/06/2022

.....

Date

.....

ZHAO YANG

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

30/06/2022

.....

Date

.....

Asst Prof. Jun Zhao

Authorship Attribution Statement

This thesis contains material from 4 papers published in the following peer-reviewed journal(s)/from papers accepted at conferences in which I am listed as an author.

Chapter 3 is published as Yang Zhao, Jun Zhao, Mengmeng Yang, Teng Wang, Ning Wang, Lingjuan Lyu, Dusit Niyato, and Kwok-Yan Lam. "Local Differential Privacy Based Federated Learning for Internet of Things," in *IEEE Internet of Things Journal, DOI: 10.1109/JIOT.2020.3037194, 2020.*

The contributions of the co-authors are as follows:

- I completed the initial draft of the manuscript.
- A/Prof. Jun Zhao provided the initial idea and edited the mathematical proofs and the manuscript drafts.
- Dr. Teng Wang helped the mathematical proof for Three-Outputs mechanism.
- Dr. Ning Wang provided the baseline codes for the experiments.
- Dr. Mengmeng Yang, Dr. Lingjuan Lyu, Prof. Dusit Niyato, and Prof. Kwoh-Yan Lam reviewed and edited the manuscript.

Chapter 4 is published as Yang Zhao, Jun Zhao, Jiawen Kang, Zehang Zhang, Dusit Niyato, Shuyu Shi, and Kwok-Yan Lam. "A Blockchain-Based Approach for Saving and Tracking Differential-Privacy Cost," in *IEEE Internet of Things Journal, DOI: 10.1109/JIOT.2021.3058209, 2020.*

The contributions of the co-authors are as follows:

- I completed the initial draft of the manuscript.
- A/Prof. Jun Zhao provided the idea and proposed the algorithm.
- Zehang Zhang helped the experiment.
- Dr. Jiawen Kang, Dr. Shuyu Shi, Prof. Dusit Niyato, and Prof. Kwoh-Yan Lam reviewed and edited the manuscript.

Leong Mei Han, Yang Zhao, and Jun Zhao. "POSTER: Blockchain-Based Differential Privacy Cost Management System." *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020.*

The contributions of the co-authors are as follows:

- Leong Mei Han and Yang Zhao completed the draft of the manuscript.

- Yang Zhao and Leong Mei Han completed the implementation of the algorithm.
- A/Prof. Jun Zhao reviewed and edited the manuscript.

Chapter 5 is published as Yang Zhao, Jun Zhao, Linshan Jiang, Rui Tan, Dusit Niyato, Zengxiang Li, Lingjuan Lyu, and Yingbo Liu. "Privacy-Preserving Blockchain-Based Federated Learning for IoT Devices," in *IEEE Internet of Things Journal, DOI: 10.1109/JIOT.2020.3017377, 2021.*

The contributions of the authors are as follows :

- I provided the idea and initialized draft of the manuscript.
- A/Prof. Jun Zhao helped mathematical proofs.
- Dr. Linshan Jiang and Prof. Rui Tan helped the experiments.
- Dr. Zengxiang Li, Prof. Dusit Niyato, Dr. Lingjuan Lyu, and Dr. Yingbo Liu reviewed and edited the manuscript.

30/06/2022 .....

Date

..... ZHAO YANG

Acknowledgements

First, I would like to express my sincerest gratitude to my supervisors, Asst Prof. Jun Zhao and Prof. Dusit Niyato, whose continuous guidance, support, and encouragement have been invaluable throughout this study. Under their close mentorship, I have successfully completed a number of publications. I have also come to know how to explore new research areas, choose research topics, build new approaches and do experiments. In particular, their enthusiasm and awareness of research have encouraged me to do better successes. Furthermore, I would like to express my thankfulness to my thesis advisory committee members Prof. NG Wee Keong, Asst Prof. Hung Dinh Nguyen, and Asst Prof. Han Yu, for their valuable comments on my research.

I would like to show my appreciation to Prof. Kwok-Yan Lam, Prof. Rui Tan, Linshan Jiang, Dr. Jiawen Kang, Dr. Mengmeng Yang, Dr. Teng Wang, Dr. Ning Wang, Dr. Shanhan Feng, Dr. Yingbo Liu, Dr. Wenbo Wang, Dr. Yue Xiu, and Dr. Wenchao Zhai. They have helped me by giving precious suggestions and directed me during my PhD days. Besides, I also would like to acknowledge my teammates, friends, and colleagues, including Muhammad Baqer Mollah, Dr. Yang Liu, Dr. Huimei Han, Chaoyu Dong, Bin Tang, Tao Bai, Yidong Lu, Chang Liu, and Xinyu Zhou, for having nice and enjoyable discussions with me.

Finally, I would like to express my deepest gratitude to my parents for their continued support.

Abstract

Massive volumes of sensitive information are being collected for data analytics and machine learning, such as large-scale Internet of Things (IoT) data. Some IoT data contain users' confidential information, for example, energy consumption or location data. These data may expose a family's habits and routines that attackers may utilize to perform attacks [1–5]. The Internet of Vehicles (IoV), a promising branch of IoT, simulates a large variety of crowdsourcing applications such as Waze, Uber, and Amazon Mechanical Turk. These applications report the real-time traffic information to the cloud server, which trains a machine learning model based on traffic information uploaded by intelligent traffic management users. However, crowdsourcing application owners can easily infer users' location information, traffic information, motor vehicle information, and environmental information, etc., raising severe sensitive personal information privacy concerns. Besides, as the number of vehicles increases, the frequent communication between vehicles and the cloud server incurs a tremendous communication cost.

Many countries have strict policies, regulations, and laws on how technology companies collect and process users' data to protect personal privacy. These companies need to analyze users' data to improve their service quality. In order to preserve privacy while revealing useful information about datasets, differential privacy (DP) is proposed [6–8]. Intuitively, the output of a DP mechanism will not change significantly because of the presence or absence of one tuple of a dataset. DP has attracted much interest from both the academia [9–13] and the industry [14–16]. For example, Apple has incorporated DP into its mobile operating system iOS [14]; Google has implemented a DP tool called RAPPOR in the Chrome browser to collect information [15]. An increasing amount of users' sensitive information is now being collected for analytic purposes. Also, DP has been widely studied in the literature to protect the privacy of users' information. The privacy parameters bound the information about the dataset leaked by the noisy output. Oftentimes, a dataset needs to be used for answering multiple queries, so the level of privacy

protection may degrade as more queries are answered. Thus, it is crucial to keep track of privacy budget spending, which should not exceed the given limit of the privacy budget. In particular, we have made the following three major contributions.

The first contribution is to integrate federated learning (FL) and local differential privacy (LDP) to facilitate the crowdsourcing applications to obtain the machine learning model to avoid the privacy leakage threat and reduce the communication cost. Specifically, we propose four LDP mechanisms to perturb gradients. The proposed `Three-Outputs` mechanism introduces three different output possibilities to deliver a high accuracy when the privacy budget is small. The output possibilities of `Three-Outputs` can be encoded with two bits to reduce the communication cost. Additionally, to maximize the performance when the privacy budget is significant, an optimal piecewise mechanism (`PM-OPT`) is proposed. We further propose a suboptimal piecewise mechanism (`PM-SUB`) with a more straightforward formula and comparable utility to the `PM-OPT` mechanism. Then, we build a novel hybrid mechanism by combining `Three-Outputs` and `PM-SUB` mechanisms. Finally, an LDP based FL stochastic gradient descent (`LDP-FedSGD`) algorithm is proposed to coordinate the cloud server and edge devices to train the machine learning model collaboratively. Applying our proposed LDP algorithms to FL protects private personal information in case adversaries infer sensitive information by reversing engineering uploaded gradients. Also, our proposed LDP algorithms ensure the utility of the gradients for FL.

The second contribution is that when a query has been answered before and is asked again on the same dataset, we may reuse the previous noisy response to answer the current query to save the privacy cost. In view of the above, we design an algorithm to reuse previous noisy responses if the same query is asked repeatedly. In particular, considering that different requests of the same query may have different DP requirements, our algorithm sets the optimal fraction from the old noisy responses to reuse and add new noise to minimize the accumulated privacy cost. In order to implement the algorithm, we design and implement a blockchain-based system for tracking and saving DP costs as the bockchain provides a distributed immutable ledger that records each query's type, the noisy response used to answer each query, the associated noise level added to the true query result, and the remaining privacy budget in our system. As a result, the dataset owner knows

how the dataset has been used and be confident that no new privacy cost will be incurred for answering queries once the specified privacy budget is exhausted.

The third contribution is to design an FL system leveraging a reputation mechanism to assist home appliance manufacturers in training a machine learning model based on customers' data to help manufacturers develop a smart home system. Then, manufacturers can predict customers' requirements and consumption behaviors in the future. The working flow of the system includes two stages: in the first stage, customers train the initial model provided by the manufacturer using both the mobile phone and the mobile edge computing (MEC) server. Customers collect data from various home appliances using phones, and then they download and train the initial model with their local data. After deriving local models, customers sign on their models and send them to the blockchain. If customers or manufacturers are malicious, we use the blockchain to replace the centralized aggregator in the traditional FL system. Since records on the blockchain are untampered, malicious customers' or manufacturers' activities are traceable. In the second stage, manufacturers select customers or organizations as miners for calculating the averaged model using received models from customers. By the end of the crowdsourcing task, one of the miners chosen as the temporary leader uploads the model to the blockchain. We enforce DP on the extracted features and propose a new normalization technique to protect customers' privacy and improve test accuracy. We experimentally demonstrate that our normalization technique outperforms batch normalization when features are under DP protection. In addition, to attract more customers to participate in the crowdsourcing FL task, we design an incentive mechanism to award participants.

In summary, this thesis addresses challenging problems faced while conducting privacy-preserving analysis on the data from IoT devices, including designing algorithms to preserve data privacy, managing the differential privacy cost wisely with blockchain, and proposing a normalization technique to improve the accuracy of the FL model. Also, we do extensive experiments by employing publicly available real datasets to confirm that our proposed algorithms and systems are valid. Finally, we list several promising research directions for future work.

Contents

[1]The work in this chapter has been published as Yang Zhao, Jun Zhao, Mengmeng Yang, Teng Wang, Ning Wang, Lingjuan Lyu, Dusit Niyato, and Kwok-Yan Lam. "Local Differential Privacy Based Federated Learning for Internet of Things", in *IEEE Internet of Things Journal, DOI: 10.1109/JIOT.2020.3037194, 2020.*

[2]The work in this chapter has been published as Yang Zhao, Jun Zhao, Jiawen Kang, Zehang Zhang, Dusit Niyato, Shuyu Shi, and Kwok-Yan Lam. "A Blockchain-Based Approach for Saving and Tracking Differential-Privacy Cost", in *IEEE Internet of Things Journal, DOI: 10.1109/JIOT.2021.3058209, 2020.*

[3]Leong Mei Han, Yang Zhao, and Jun Zhao. "POSTER: Blockchain-Based Differential Privacy Cost Management System." *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020.*

[4]The work in this chapter has been published as Yang Zhao, Jun Zhao, Linshan Jiang, Rui Tan, Dusit Niyato, Zengxiang Li, Lingjuan Lyu, and Yingbo Liu. "Privacy-Preserving Blockchain-Based Federated Learning for IoT Devices", in *IEEE Internet of Things Journal, DOI: 10.1109/JIOT.2020.3017377, 2021.*

List of Figures

List of Tables

Symbols and Acronyms

Acronyms

AI	Artificial Intelligence
IoT	Internet of Things
FL	Federated Learning
DP	Differential Privacy
LDP	Local Differential Privacy
SGD	Stochastic Gradient Descent
PM	Piecewise Mechanism
HM	Hybrid Mechanism
ULDP	User-Level Privacy Protection with Distributed Perturbation
RLDP	Record-Level Privacy Protection with Distributed Perturbation
RLCP	Record-Level Privacy Protection with Centralized Perturbation
ULCP	User-Level Privacy Protection with Centralized Perturbation
IoV	Internet of Vehicles
MEC	Mobile Edge Computing
PM-OPT	Optimal Piecewise Mechanism
PM-SUB	Suboptimal Piecewise Mechanism
LDP-FedSGD	LDP based FL stochastic gradient descent

Chapter 1

Introduction

In this chapter, the background and motivation of applying privacy technologies such as differential privacy (DP), local differential privacy (LDP), and federated learning (FL) to preserve the privacy of data generated by the Internet of Things (IoT) devices are introduced. Then, contributions are summarized, and the outline for the rest of the thesis is presented.

1.1 Background

1.1.1 Background of IoT

Kevin Ashton coined the concept of IoT in 1999 [20]. IoT is a network formed by connected things; things represent objects with sensors and computing capability. With the proliferation of wireless communication technologies and hardware techniques, billions of devices are interconnected in various areas such as healthcare, transportation, manufacturing, and home appliances. Due to the booming increase of devices, data generated by them are enormous. These data help IoT companies learn more about clients' lifestyles, eventually improving clients' quality of life. However, data may contain clients' sensitive and confidential information; thus, many companies and clients are unwilling to disclose clients' data. For example, wearable devices may monitor and collect clients' health status, which may seriously comprise clients' privacy; the development of sensors and communication technologies for IoT enables a fast and large-scale collection of data, which has

bred new services and applications such as the Waze application that provides the intelligent transportation routing service. Specifically, Waze aims at providing the real-time traffic navigation service [21–23]. The Waze application crowdsources real-time road and traffic data from users of Waze to provide them improved traffic service. For instance, the Waze application suggests drivers suitable routes to help drivers to avoid congestion. This kind of service benefits users' daily life, but it may raise privacy concerns of sensitive data such as users' health data in the wearable devices and users' location information [24–27]. Besides, as the number of IoT devices increases, IoT smart devices usually generate tremendous data. In the same connection, it might be impractical if all data are sent to the centralized server for analysis. This is because of concerns about the network's bandwidth limitation, communication costs, and privacy leakage. Therefore, more privacy-preserving techniques are urgently needed.

1.1.2 Background of Privacy

In recent years, artificial intelligence (AI) becomes more and more popular. Machine learning is an AI algorithm to learn and derive patterns from data with little human intervention. The derived patterns can be used to make predictions for future decisions. Deep learning is the most widely used machine learning algorithms nowadays. The explosive increase of the data volume contributes to the ever-growing number of deep learning applications. In addition, the development of hardware technologies (e.g., GPUs) also facilitates the processing of deep learning algorithms. We witness the explosive growth of deep learning applications in various areas such as computer vision, speech recognition, and recommender systems. Since many deep learning algorithms are data-driven, research institutes and companies depend on collected data for building advanced deep learning models to improve their business. As data are crowdsourced from users' devices, data may be sensitive. Therefore, privacy-preserving techniques like DP, LDP, and FL are utilized. In the following, we introduce them respectively in detail.

1.1.2.1 Differential Privacy

Statistical analysis has been widely used by various healthcare, finance, and service organizations, etc., for many years. Organizations try to improve their products

and services by analyzing clients' data and feedback. Traditionally, a common approach to protect users' privacy is to anonymize part of sensitive information (e.g., names and IDs) to protect data privacy. But Dwork *et al.* [8] have demonstrated that de-anonymization of the sensitive information is insufficient in protecting the privacy of the data because the rest of personal information can still be used to find the exact identities. By checking shared attributes in other datasets, attackers can re-identify the unique individual. For instance, Netflix's DVD rental service held a data analytic contest to improve the movie commendation service in 2006. They published a dataset including $100,480,507$ movie ratings voted by $480,189$ subscribers. The approach they used to protect users' privacy is to de-anonymize users' personal information and left with ratings and movies. However, Narayanan and Shmatikov [28] re-identified users by cross-checking movie ratings in the Internet Movie Database (IMDb) [29]; Massachusetts' Group Insurance Commission publicized 135,000 state employees' medical records after deleting sensitive personal information such as addresses and names in the 1990s. But later, Sweeney [30] re-identified those patients by correlating with public voting records using shared attributes, for instance, birth dates and zip codes. Since many deep learning algorithms are data-driven, research institutes and companies seek to use collected data to build advanced deep learning models to improve their business. Because data are crowdsourced from users, users' sensitive information may also be included.

To protect the privacy of data while conducting statistical analysis or deep learning, Dwork *et al.* [8] proposed the definition of DP. DP ensures that the adversaries cannot determine with high confidence whether the randomized output comes from a dataset D or its neighboring dataset D', which differs from D by one record. Thus, it guarantees that individual data do not affect the publicity of the whole dataset, and attackers cannot infer any personal data from the released differentially private results.

1.1.2.2 Local Differential Privacy

One drawback of centralized DP is that it requires data owners to trust the central authority. Data owners send their raw data to the central authority for adding noises. However, the central authority is not always trustful, because they may leak users' information or inappropriately use data. LDP was proposed by [31] to

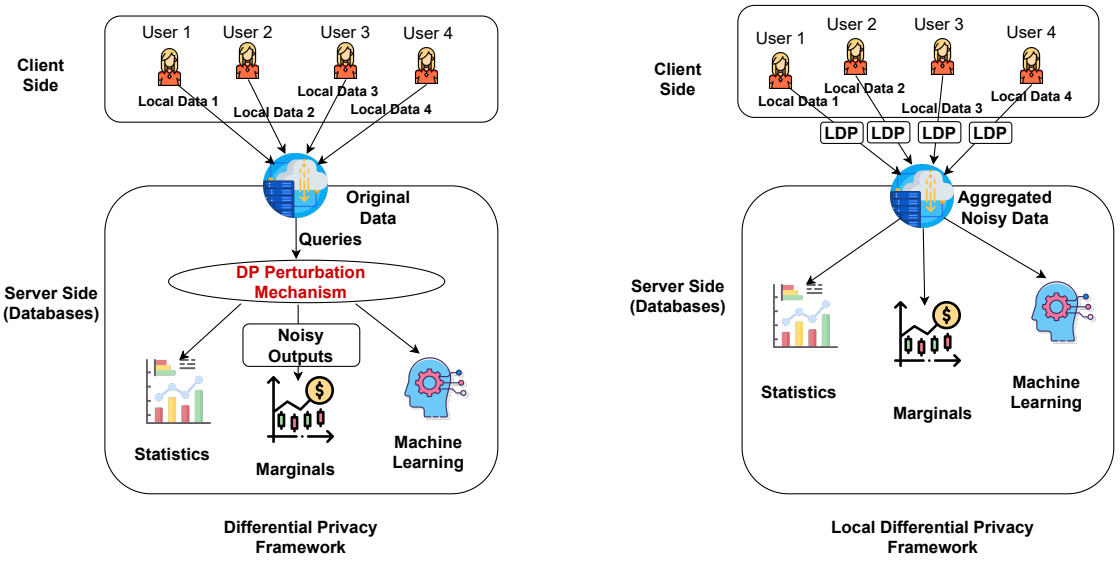


FIGURE 1.1: Differential Privacy Framework vs Local Differential Privacy Framework.

prevent the malicious third-party central authority. Figure 1.1 illustrates the DP framework and LDP framework.

In particular, LDP is more rigorous than the centralized DP. In LDP, users complete the perturbation by themselves. Each user runs a random perturbation algorithm with the local dataset, and then the user sends perturbed results to the aggregator. The privacy budget ϵ controls the trade-off between privacy and utility, and a higher privacy budget means lower privacy protection, whereas a lower privacy budget represents higher privacy protection. LDP guarantees that even if the attacker has the access to the individual's response to a query, the attacker cannot learn anything useful from the noisy response.

1.1.2.3 Federated Learning

Aside from concerns on data privacy, traditional machine learning algorithms may have some flaws in preserving privacy and saving communication costs. First, since traditional machine learning algorithms follow a centralized training style, training data are stored in a central server for further processing and training. If a third-party server leaks the information, the privacy of users' data may be compromised. Besides, many countries launch laws to preserve data privacy; for example, the Consumer Privacy Bill of Rights in the US and the General Data Protection Regulation (GDPR) in Europe are launched, aiming to protect individual privacy.

Second, with the number of IoT devices increasing dramatically, a large sum of data is generated daily. Since IoT devices are decentralized, transmitting a large sum of data to cloud servers may drain the bandwidth of the communication system. IoT devices may also participate in machine learning. Finally, as the IoT devices' processing capability improves, edge devices gain intelligence and train machine learning models locally. It is unnecessary to leverage the remote server for training. Along with the motivations above, McMahan *et al.* [32, 33] from Google proposed the concept of federated learning.

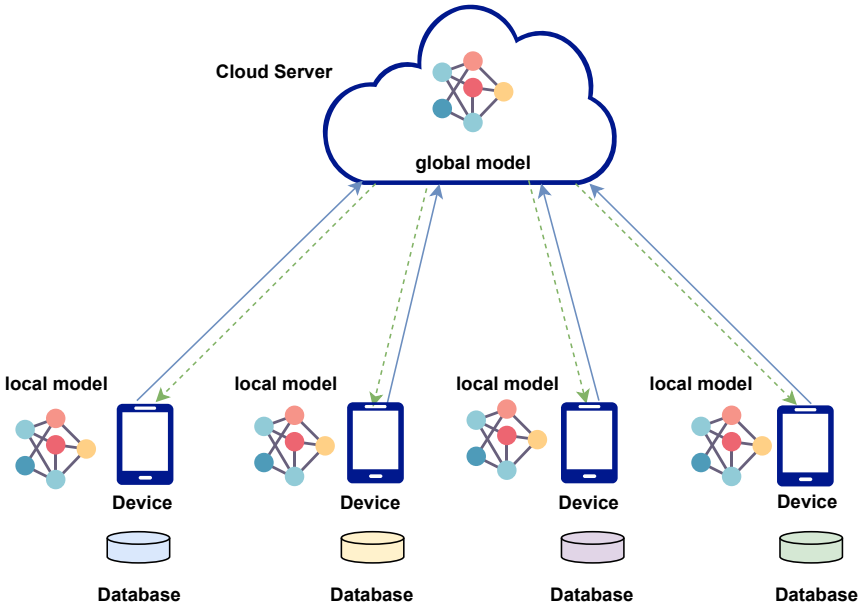


FIGURE 1.2: Federated Learning.

Figure 1.2 illustrates the working flow of FL. In FL, data are distributed and scattered among different clients, and no such single node stores a whole dataset [34, 35]. Specifically, FL is a distributed machine learning paradigm that trains machine learning models while keeping data decentralized. Raw data remain local at participants' and will not be transferred to the central server. Participants of FL train machine learning models locally, and subsequently, they send their locally trained models to a centralized server to aggregate.

FL is similar to the traditional distributed machine learning [36, 37], but the assumptions of local datasets are different. More specifically, traditional distributed learning aims at optimizing the parallel computing power. Still, data are IID among various parties, while FL focuses on the heterogeneous local datasets, meaning that

training data can be distributed, non-IID, and unbalanced among various participants. Each participant trains the same initial model using their local data to collaboratively obtain a global model with the minimized averaged sum of loss functions among all participants. In particular, the workflow of FL is that each user does the initial model training by utilizing the local dataset and then uploads the trained model instead of raw data to a central server. Then, the centralized server obtains a global model by averaging all of the uploaded models. As many decentralized nodes are building machine learning models instead of relying on a single centralized server, FL effectively prevents the single point of failure.

Nevertheless, some challenges exist in data analytics which shall be resolved in the next section.

1.2 Research Challenges, Motivation, and Methodologies

This section lists research challenges that motivate us to conduct the research studies in this thesis.

1.2.1 Impacts of Privacy-preserving Technologies on Machine Learning

The privacy-preserving machine learning algorithm using data from distributed IoT devices is challenging. In order to resolve challenges, existing solutions leverage FL technologies. FL enables analysts to analyze and utilize the locally generated data in a decentralized way without requiring uploading data to a centralized server; that is, the utility of data is well maintained despite data are preserved locally. But an honest-but-curious aggregator may be able to leverage users' uploaded gradients to infer the original data [38, 39]. To further address the aforementioned privacy issues in FL models, we leverage LDP mechanisms to protect the machine learning models' privacy. Existing LDP mechanisms include three popular LDP mechanisms which are `Duchi` *et al.*'s solution [17], `Laplace` mechanism [7], and Piecewise Mechanism (`PM`) [18].

TABLE 1.1: A comparison of the worst-case variances of existing LDP mechanisms on a single numeric attribute with a domain $[-1, 1]$: `Duchi` of [17] generating a binary output, `Laplace` of [7] with the addition of Laplace noise, and the `PM` of [18]. For an LDP mechanism $\mathcal{M}$, its worst-case variance is denoted by $V_{\mathcal{M}}$ and ϵ denotes the privacy budget. We obtain this table based on results of [18].

Range of ϵ	Comparison of mechanisms
$0 < \epsilon < 1.29$	$V_{\texttt{Duchi}} < V_{\texttt{PM}} < V_{\texttt{Laplace}}$
$1.29 < \epsilon < 2.32$	$V_{\texttt{PM}} < V_{\texttt{Duchi}} < V_{\texttt{Laplace}}$
$\epsilon > 2.32$	$V_{\texttt{PM}} < V_{\texttt{Laplace}} < V_{\texttt{Duchi}}$

Results of [18] show that among the above three LDP mechanisms, in terms of the worst-case variance as presented in Table 1.1, `Duchi`'s solution is the best when the privacy budget is in the range of $0 < \epsilon < 1.29$, while `PM` is the best for the privacy budget $\epsilon > 1.29$. Then, a natural research question is that can we propose better or even optimal LDP mechanisms ? We would like to find a mechanism that can improve the utility of existing mechanisms.

In Chapter 3, we propose solutions to address the above challenge. We derive the optimal mechanism `PM-OPT` under the "piecewise framework" of [18]. Since expressions for `PM-OPT` are quite complex, we present `PM-SUB` which is suboptimal, whereas it has simpler expressions and achieves a comparable utility compared with `PM-OPT`. Also, we propose a `Three-Outputs` mechanism, which contains three output possibilities such that it can reduce communication costs. In addition, we propose a hybrid mechanism that takes advantage of `Three-Outputs` and `PM-SUB` to obtain a better utility and smaller worst-case variance. Moreover, we integrate FL [32] with LDP [40] techniques. FL facilitates collaborative learning with uploaded gradients from users instead of sharing users' raw data. Adding LDP noises to gradients before uploading, the LDP based FL framework prevents attackers from deducing original data even though attackers obtain perturbed gradients.

1.2.2 Impacts of Blockchain on Differential-Privacy Cost Management

A randomized mechanism satisfies (ϵ, δ)-DP [6], if, for any two adjacent databases, the output cannot tell them apart by more than a multiplicative factor e^{ϵ}, except

with a probability δ of information accidentally being leaked. Thus, the information about the dataset leaked by the noisy output of an (ϵ, δ)-DP mechanism is bounded by the privacy parameters ϵ and δ. Smaller ϵ and δ mean more robust privacy protection and less information leakage. Note that non-zero information leakage is necessary to achieve non-zero utility. Usually, a dataset may be used for answering multiple queries (e.g., for various analytic tasks), thus accumulating the information leakage and degrading the privacy protection level, which can be intuitively understood as the increase of private spending. Therefore, it is necessary to record the privacy cost to prevent it from exceeding the privacy budget. When a differentially private mechanism is applied to real-world applications, the privacy budget is utilized to quantify the risk of privacy leakage. Besides, we reduce privacy costs by reusing old noisy responses to answer the current query if the query was answered before.

Traditionally, the privacy cost incurred by answering queries on a dataset is claimed by the dataset holder. Users whose information is in the dataset are not clear about the usage. Privacy consumption may have exceeded the privacy budget. Managing privacy costs may use the solution inspired by the emerging blockchain technology to resolve the issue. Blockchain is a chain of blocks storing cryptographic and tamper-resistant transaction records without using a centralized server [41, 42]. With blockchain recording how the dataset is utilized in answering queries, users fully know how their information is analyzed. Users can easily access the blockchain to check the consumption of the privacy budget. The dataset holder has the motivation to adopt our blockchain-based approach to provide the following accountability guarantee to users whose information is in the dataset: if the dataset holder uses the dataset more than the set of queries recorded by the blockchain, measures can be taken to catch the dataset holder with cheating because transactions written into the blockchain are tamper-resistant. Yang *et al.* [43] proposed to leverage blockchain to track the DP budget, but they did not propose a mechanism to reuse noise. In contrast, we design a DP mechanism to effectively utilize previous queries' results to reuse noise and reduce privacy costs.

In Chapter 4, we resolve the above challenges by proposing a blockchain-based DP algorithm to track and manage the differential-privacy cost, which uses blockchain to make the privacy spending transparent to the data owner. Consequently, the data owner can track how the dataset is used by checking blockchain transactions'

information, including each query's type, the noisy response used to answer each query, the associated noise level added to the actual query result, and the remaining privacy budget. In addition to providing transparency of privacy management, another advantage of our blockchain-based system is as follows. Once the specified privacy budget is exhausted, a smart contract implemented on the blockchain ensures that no new privacy cost will be incurred, and this can be verified. Furthermore, since the blockchain stores the noisy response used to answer each query, we also design an algorithm to minimize the accumulated privacy costs by reusing previous noisy responses if the same query is asked again. Our algorithm can set the optimal reuse fraction of the old noisy response and add new noise considering different requests of the same query may be sent with varying privacy requirements. In our blockchain-based system, reusing noisy responses not only saves privacy costs but also reduces communication overhead when the noisy response is generated without contacting the server hosting the dataset.

1.2.3 Impacts of Differential Privacy and Blockchain on Federated Learning

With the proliferation of smart home devices, tremendous data are generated. FL enables analysts to analyze and utilize the locally generated data in a decentralized way without requiring uploading data to a centralized server; that is, the utility of data is well maintained despite data are preserved locally. We design an FL-based system to help home appliance manufacturers use data generated in customers' appliances smartly and conveniently. Our system considers home appliances of the same brand in a family as a unit, and a mobile phone is used to collect data from home appliances periodically and train the machine learning model locally [44]. Since mobile phones have limited computational power and battery life, we offload the training task to the edge computing server. Then, the blockchain smart contract is leveraged to generate a global model by averaging the sum of locally trained models submitted by users. In this federated way, source data are supposed to maintain security and privacy.

However, Melis *et al.* [45] demonstrated that gradient updates might leak significant information about customers' training data. Attackers can recover data from gradients uploaded by customers [38]. Besides, the federated approach for training

the model is susceptible to model poisoning attacks [46]. In addition, information leakage risks exist in the third party's mobile edge computing (MEC) server [47]. To address the aforementioned security and privacy issues, we adopt blockchain and DP. It is worth noting that Apple has successfully applied DP in FL to improve the privacy of its popular voice assistant service Siri [48]. Specifically, manufacturers upload a preliminary model with initialized parameters. The model is available on the blockchain for customers to download and train with their local data. The blockchain assists the crowdsourcing requester (i.e., manufacturer) audit whether there are malicious updates from customers. The traditional crowdsourcing system is hosted by a third party, which charges customers costly service fees, while our designed system uses blockchain to record crowdsourcing activities. Therefore, customers and the requester can save high service fees while keeping the crowdsourcing system functional. Due to the limitation of the block size, we propose to use the InterPlanetary File System (IPFS) [49] as the distributed storage solution when the model size is large.

1.3 Summary of Contributions and Outline of the Thesis

Our contributions are summarized as follows:

- In Chapter 1, we present the background of this thesis. In particular, we briefly introduce IoT, DP, LDP, and FL and research challenges, motivations, and methodologies.
- In Chapter 2, the literature review for existing studies that are related to our research is presented. In addition, we compare existing solutions and highlight the novelty of our proposed approaches.
- In Chapter 3, we investigate the solutions for protecting privacy while analyzing numerical data. By proposing the `LDP-FedSGD` mechanism for FL in IoV, we present novel LDP mechanisms for numeric data with a continuous domain. Among our proposed mechanisms, `Three-Outputs` and `PM-SUB` outperform existing mechanisms for a wide range of DP budget ϵ and confirmed by experiments. In terms of comparing our `Three-Outputs` and `PM-SUB`,

we have: `Three-Outputs`, whose output includes three possibilities, achieves a smaller worst-case noisy variance when ϵ is small. In contrast, `PM-SUB`, whose output can take infinite possibilities of an interval, has higher a smaller worst-case noisy variance when ϵ is large. Our `PM-SUB` is a slightly suboptimal version of our `PM-OPT` to simplify the expressions. We further combine `Three-Outputs` and `PM-SUB` to obtain a hybrid mechanism `HM-TP`, which achieves even higher utility.

- In Chapter 4, a novel privacy-preserving algorithm with rigorous mathematical proof is designed to minimize accumulated privacy costs under a limited privacy budget by reusing previous noisy responses if the same query is received. Thus, a dataset can answer more queries while preventing privacy leakage, which is essential for the datasets with frequent queries, e.g., medical record datasets. In addition, we implement the algorithm using blockchain to make privacy spending transparent to the data owner. Consequently, the data owner can track how the dataset is used by checking blockchain transactions' information, including each query's type, the noisy response used to answer each query, the associated noise level added to the actual query result, and the remaining privacy budget.

- In Chapter 5, we design an FL-based system to help home appliance manufacturers smartly and conveniently use data generated in customers' appliances. Our system considers home appliances of the same brand in a family as a unit, and a mobile phone is used to collect data from home appliances periodically and train the machine learning model locally [44]. Since mobile phones have limited computational power and battery life, we offload the training task to the edge computing server. Then, the blockchain smart contract is leveraged to generate a global model by averaging the sum of locally trained models submitted by users. In this federated way, source data are supposed to maintain security and privacy. Customers extract data features in the mobile using the deployed feature extractor and add noise with a formal privacy guarantee to perturb the extracted features in the first step. In the second step, customers train fully connected model layers with perturbed features in the MEC server. Moreover, we improve the traditional batch normalization by removing constraints of mean value and variance while constraining the bound within $[-\sqrt{N-1}, \sqrt{N-1}]$, where N denotes the batch size. After training, customers sign on hashes of encrypted models with their private

keys and transmit locally trained models to the blockchain. Selected miners verify the identities of senders, download models, and calculate the average of all model parameters to obtain the global model. One miner, chosen as the temporary leader, encrypts and uploads the global model to the blockchain.

- In Chapter 6, we conclude the thesis and provide several research directions for future work.
- Appendix A and Appendix B present the mathematical proofs for Chapter 3 and Chapter 4, respectively.

Chapter 2

Literature Review

In this chapter, we introduce concepts that will be used in the rest of the thesis. We introduce formal definitions of differential privacy (DP), local differential privacy (LDP), federated learning (FL), and blockchain in Section 2.1, Section 2.2, Section 2.3, and Section 2.4. Also, we review prior work related to privacy-preserving data analysis. Then, we highlight novel contributions of each chapter and their significance compared with existing solutions.

2.1 Differential Privacy

Differential privacy is a mathematical model to guarantee the database's privacy. In the following, we introduce the formal definition and existing studies of DP.

2.1.1 Differential Privacy Definition

Let D be a dataset that collects data from data universe χ and contains n individuals. That is, the dataset $D \in \chi^n$. The formal definition of (ϵ, δ)-*differential privacy* is as follows:

Definition 2.1 ((ϵ, δ)**-Differential Privacy** [8]**).** A randomized mechanism $\mathcal{M}$ provides (ϵ, δ)-*differential privacy* if any two neighbouring datasets D and D' (i.e.,

D and D' differ in at most one record), $\mathcal{M}$ guarantees that

$$\mathbb{P}\left[\mathcal{M}(D) \in Y\right] \leq e^{\epsilon} \cdot \mathbb{P}\left[\mathcal{M}(D') \in Y\right] + \delta,$$

where $\mathbb{P}\left[\cdot\right]$ denotes the probability, and the probability space is over the coin flips of the randomized mechanism $\mathcal{M}$. Y iterates through all subsets of the output range of mechanism $\mathcal{M}$.

Definition 2.2 (**Neighboring Datasets**)**.** If datasets D and D' differ only in one tuple, they are called neighbouring. It includes two variants: First, the sizes of datasets D and D' differ in one, meaning that D' is achieved by adding or deleting one tuple from D. Second, the sizes of datasets D and D' are the same, but only one of their tuples is different. The notion of neighbouring datasets includes both of the above two cases.

Remark 2.1. The notion of (ϵ, δ)-*differential privacy* under $\delta = 0$ becomes ϵ-*differential privacy*. ϵ-*differential privacy* and (ϵ, δ)-differential privacy are called as *pure* and *approximate* differential privacy, respectively, in many studies [9–11].

2.1.2 Existing Differential Privacy Mechanisms

In this section, we introduce two popularly used DP mechanisms, including `Laplace` mechanism and `Gaussian` mechanism.

2.1.2.1 The Laplace Mechanism

The `Laplace` mechanism of [7] can be used to ensure differential privacy by adding the independent zero-mean Laplace noise with scale λ to each dimension of the output. Specifically, λ equals $\Delta f/\epsilon$, where Δf is the ℓ_1-sensitivity of the query f and it measures the maximum change of the true query output over neighboring databases. A large Δf indicates that f may leak a tuple's vital information, so that a large amount of noise is required to be injected to the output of f to protect privacy. The formal definition of the ℓ_1-sensitivity is as follows:

Definition 2.3 (ℓ_1**-sensitivity**)**.** The ℓ_1-sensitivity is also called the global sensitivity. D and D' are neighboring datasets. Let function $f : \chi^n \to \mathbb{R}^k$. Then, the

ℓ_1-sensitivity of function f is:

$$\Delta f = \max_{\text{neighboring datasets } D, D'} \|f(D) - f(D')\|_1. \tag{2.1}$$

Definition 2.4 (**The Laplace Distribution**)**.** The Laplace distribution is the distribution with probability density function:

$$Lap(x|\lambda) = \frac{1}{2\lambda} exp\left(-\frac{\|x-\theta\|_1}{\lambda}\right), \tag{2.2}$$

where λ represents the scale, and θ is the location which equals to 0 meaning that the center is at 0. We use $Lap(\lambda)$ to represent the Laplace distribution with scale λ and denote a random variable $X \sim Lap(\lambda)$.

The definition of the `Laplace` mechanism is as follows:

Definition 2.5 (**The Laplace Mechanism**)**.** Given a query $f : \chi^n \to \mathbb{R}^k$, the definition of the `Laplace` mechanism is as follows:

$$\mathcal{M}(D, f, \epsilon) = f(D) + (Y_1, \cdots, Y_k), \tag{2.3}$$

where Y_i are i.i.d. random variables drawn from $Lap(\Delta f/\epsilon)$.

In addition to the `Laplace` mechanism, the `Gaussian` mechanism proposed in [6] is popularly utilized. The improved result given by [8] is Lemma 2.1.

2.1.2.2 The Gaussian Mechanism

The `Gaussian` mechanism uses the normally distributed noise to ensure DP. In particular, the `Gaussian` mechanism will return true query response $Q(D)$ plus noise η, i.e., $Q(D) + \eta$ where the noise η follows the normal distribution.

Definition 2.6 (ℓ_2**-sensitivity** [8])**.** A query Q's ℓ_2-sensitivity for any neighboring datasets D and D' is defined as

$$\Delta_Q = \max_{\text{neighboring datasets } D, D'} \|Q(D) - Q(D')\|_2,$$

i.e., the maximal ℓ_2 distance between the real query results. For one-dimensional real-valued query Q, Δ_Q is simply the maximal absolute difference between $Q(D)$ and $Q(D')$ for any neighboring datasets D and D'.

Lemma 2.1 (**Theorem A.1 by Dwork and Roth [8]**). *To answer a query Q with ℓ_2-sensitivity Δ_Q, for any $\epsilon, \delta \in (0, 1)$, adding a zero-mean Gaussian noise with standard deviation $\sqrt{2 \ln \frac{1.25}{\delta}} \times \frac{\Delta_Q}{\epsilon}$ (denoted by* Gaussian$(\Delta_Q, \epsilon, \delta)$ *hereafter in Chapter 4) to each dimension of the true query result achieves (ϵ, δ)-differential privacy.*

2.1.3 Existing Studies on Differential Privacy Mechanisms

Inspired by the `Laplace` mechanism 2.1.2.1 and `Gaussian` mechanism 2.1.2.2, more differential privacy algorithms are proposed to provide privacy protection. For example, Xiao *et al.* [50] proposed an algorithm to correlate the Laplace noise added to different queries to improve the overall accuracy. Given a series of counting queries, the mechanism proposed by Li and Miklau [51] selected a subset of queries to answer privately and uses their noisy answers to derive answers for the remaining queries. For a set of non-overlapping counting queries, Kellaris and Papadopoulos *et al.* [52] pre-processed the counts by elaborate grouping and smoothing them via averaging to reduce the sensitivity and thus the amount of injected noise. Given a workload of queries, Yaroslavtsev *et al.* [53] introduced a solution to balance accuracy and efficiency by answering some queries more accurately than others. Above DP algorithms have addressed challenges in reducing noise to add or improve accuracy when answering queries. Nevertheless, none of them propose an efficient algorithm to reuse the previous query results or noises. Motivated by this direction, in Chapter 4, we propose a blockchain-based DP algorithm that reuses previous queries and results such that the dataset can be used to answer more queries at a limited differential privacy budget.

2.2 Local Differential Privacy

Apart from DP, LDP has attracted much attention [54–61]. LDP can be understood as a variant of DP, where the difference is the definition of "neighboring datasets".

In DP, two datasets are neighboring if they differ in just one record; in LDP, any two instances of the user's data are neighboring. LDP is formally defined as follows:

Definition 2.7 (**Local Differential Privacy**)**.** Let $\mathcal{M}$ be a randomized function with domain $\mathbb{X}$ and range $\mathbb{Y}$; i.e., $\mathcal{M}$ maps each element in $\mathbb{X}$ to a probability distribution with sample space $\mathbb{Y}$. For a non-negative ϵ, the randomized mechanism $\mathcal{M}$ satisfies ϵ-local differential privacy if

$$\left| \ln \frac{\mathbb{P}_{\mathcal{M}}[Y \in S|x]}{\mathbb{P}_{\mathcal{M}}[Y \in S|x']} \right| \leq \epsilon, \ \forall x, x' \in \mathbb{X}, \ \forall S \subseteq \mathbb{Y}, \tag{2.4}$$

where $\mathbb{P}_{\mathcal{M}}[\cdot|\cdot]$ means the conditional probability distribution depending on $\mathcal{M}$. In LDP, users are responsible for the random perturbation instead of a centralized aggregator. The centralized aggregator only receives perturbed results which make sure that the aggregator is unable to determine whether the true tuple is x or x' with high probability depending on the privacy budget ϵ.

2.2.1 Existing Local Differential Privacy Mechanisms

Recently, LDP has attracted a lot of attention [54–61]. Several mechanisms for numeric data estimation have been proposed [7, 17, 18, 62]. (i) Dwork *et al.* [7] proposed the `Laplace` mechanism, which added the Laplace noise to real one-dimensional data directly. The `Laplace` mechanism is originally used in the centralized differential privacy mechanism, and it can be applied to LDP directly. (ii) For a single numeric attribute with a domain $[-1, 1]$, `Duchi` *et al.* [17] proposed an LDP framework that provided output from $\{-\frac{e^{\epsilon}+1}{e^{\epsilon}-1}, \frac{e^{\epsilon}+1}{e^{\epsilon}-1}\}$. (iii) Wang *et al.* [18] proposed the `PM` which offered an output that contains infinite possibilities in the range of $[-\frac{e^{\epsilon/2}+1}{e^{\epsilon/2}-1}, \frac{e^{\epsilon/2}+1}{e^{\epsilon/2}-1}]$. In addition, they applied the LDP mechanism to preserve the privacy of gradients generated during machine learning tasks. Both approaches by `Duchi` *et al.* [17] and Wang *et al.* [18] can be extended to the case of multidimensional numerical data. Details of these mechanisms are introduced as follows, respectively.

2.2.1.1 The Laplace Mechanism

LDP setting also can use the `Laplace` mechanism. Specifically, the $Lap(\lambda)$ represents a random variable which obeys the Laplace distribution with scale λ. The

probability density function of the `Laplace` mechanism is defined in Eq. (2.2).

2.2.1.2 Duchi *et al.*'s Solution

`Duchi` *et al.* [17] introduced alternative mechanisms for LDP. For a single numeric attribute, one mechanism hereinafter referred to `Duchi` *et al.*'s solution of [17], flips a coin with two possibilities to generate an output, where the probability of each output depends on the input. A disadvantage of `Duchi` *et al.*'s solution is as follows: Since the output of `Duchi` *et al.*'s solution has only two possibilities, the utility may not be high for a large ϵ (intuitively, for large ϵ, the privacy protection is weak so the output should be close to the input which means the output should have many possibilities since the input can take any value in $[-1, 1]$)[1].

2.2.1.3 Piecewise Mechanism

Wang *et al.* [18] proposed the `PM` which achieves higher utility than `Duchi` *et al.*'s solution for a large ϵ, since the output range of `PM` is continuous and has infinite possibilities, instead of just 2 possibilities as in `Duchi` *et al.*'s solution. In `PM`, the plot of the output's probability density function with respect to the output value consists of three "pieces". As the input increases, the length of the leftmost (resp., rightmost) piece increases (resp., decreases), but the length of the center piece remains unchanged. Since `PM` is tailored for LDP, unlike `Laplace` for LDP, `PM` has a strictly lower worst-case variance (i.e., the maximum variance with respect to the input given ϵ) than `Laplace` for *any* ϵ.

2.2.1.4 Hybrid Mechanism

In order to fully take advantage of both `Duchi` *et al.*'s solution and `PM`, Wang *et al.* [18] developed a new Hybrid Mechanism (`HM`) by flipping a coin. With the probability α (resp. $1-\alpha$), `PM` is invoked (resp. `Duchi` *et al.*'s solution is invoked).

[1]Note that any algorithm satisfying DP or LDP has the following property: the set of possible values for the output does not depend on the input (though the output distribution depends on the input). This can be easily seen by contradiction. Suppose an output y is possible for input x but not for x' (x and x' satisfy the neighboring relation in DP or LDP). Then $\mathbb{P}[y \mid x] > 0$ and $\mathbb{P}[y \mid x'] = 0$, resulting in $\mathbb{P}[y \mid x] > e^{\epsilon}\mathbb{P}[y \mid x']$ and hence violating the privacy requirement ($\mathbb{P}[\cdot \mid \cdot]$ denotes conditional probability).

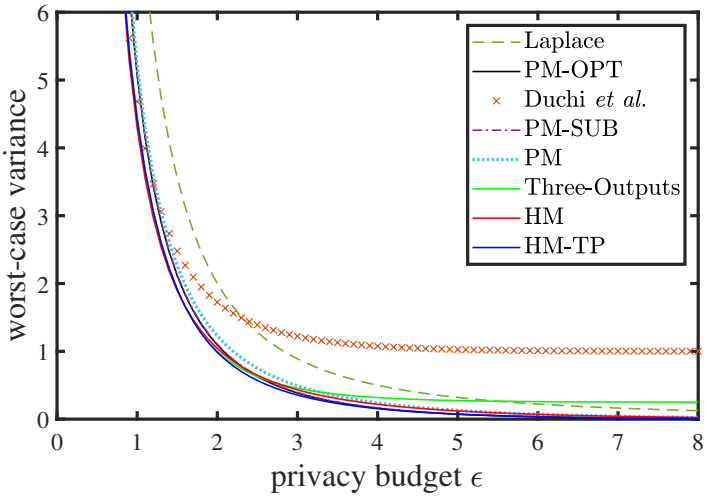


FIGURE 2.1: The worst-case of different mechanisms' noise variance for one-dimensional numeric data w.r.t. the privacy budget ϵ.

2.2.1.5 Deficiencies of Existing Local Differential Privacy Mechanisms

Fig. 2.1 illustrates that when $\epsilon \leq 2.3$, the `Laplace` mechanism's worst-case noise variance is larger than that of `Duchi` *et al.*'s [63] solution; however, the `Laplace` mechanism outperforms `Duchi` *et al.*'s [63] solution if ϵ is larger. The worst-case noise variance of `PM` is smaller than that of `Laplace` and `Duchi` *et al.*'s [63] solution when ϵ is large. The `HM` mechanism outperforms other existing solutions by taking advantage of `Duchi` *et al.*'s [63] solution when ϵ is small and `PM` when ϵ is large. However, `PM` and `HM`'s outputs have infinite possibilities that are hard to encode. We would like to find a mechanism that can improve the utility of existing mechanisms. In addition, we believe there is a mechanism that retains a high utility and is easy to encode its outputs. Based on the above intuition, in Chapter 3, we propose four novel LDP mechanisms that can be used to preserve the privacy of the data as well as prevent the privacy leakage of gradients in FL.

2.2.2 Existing Studies on Local Differential Privacy for IoT Applications

LDP has been widely used in the research of IoT [64–68]. For example, Xu *et al.* [64] integrated deep learning with LDP techniques to protect users' privacy in edge computing applications. They developed an EdgeSanitizer framework that formed

a new protection layer against sensitive inference by leveraging a deep learning model to mask the learned features with noise and minimize data. Choi *et al.* [65] explored the feasibility of applying LDP on ultra-low-power (ULP) systems. They used resampling, thresholding, and a privacy budget control algorithm to overcome the low resolution and fixed point nature of ULPs. He *et al.* [66] addressed the location privacy and usage pattern privacy induced by the MEC's wireless task offloading feature by proposing a privacy-aware task offloading scheduling algorithm based on the constrained Markov decision process. Li *et al.* [67] proposed a scheme for privacy-preserving data aggregation in the MEC to assist IoT applications with three participants, i.e., a public cloud center (PCC), an edge server (ES), and a terminal device (TD). TDs generated and encrypted data and sent them to the ES, and then the ES submitted the aggregated data to the PCC. The PCC used its private key to recover the aggregated plaintext data. Their scheme provides source authentication and integrity and guarantees the data privacy of the TDs. In addition, their scheme can save half of the communication costs. To protect the privacy of massive data generated from IoT platforms, Arachchige *et al.* [68] designed an LDP mechanism named LATENT for deep learning. A randomization layer between the convolutional and fully connected modules is added to the LATENT to perturb data before data leave data owners for machine learning services. Pihur *et al.* [69] proposed the Podium Mechanism, which is similar to our `PM-SUB`, but his mechanism applies to DP instead of LDP. Nevertheless, the above literature focuses on applying existing LDP mechanisms to various applications instead of proposing novel LDP mechanisms. On the other side, we have proposed four novel LDP mechanisms for numerical data analysis in Chapter 3.

2.3 Federated Learning

FL is a distributed machine learning paradigm, and it is used to address data privacy problems in machine learning [32, 33]. FL resolves the federated optimization problem in the distributed machine learning setting where training data are kept local on users' devices instead of gathering in a single data center [70]. In the following, we present two widely used FL algorithms, which are `FedSGD` [71] and `FedAvg` [32] algorithms:

- **Federated Stochastic Gradient Descent (FedSGD):** Recent successful deep learning applications widely adopt the stochastic gradient descent (SGD) method; thus, it is natural to apply SGD into FL algorithms. The first step of `FedSGD` is to select a fraction of the clients and distribute the current global model θ to the selected clients. Each selected client calculates the gradient using local data and sends the gradient to the server for calculating the average gradient. Next, the server uses the averaged gradient to update the global model parameters. However, it requires numerous training rounds, which is wasteful in FL since only one step of SGD is computed locally in a communication round. Due to the aforementioned premises of FL, more computation should be added in one communication round, and `FedAvg` is an efficient approach that addresses this issue.
- **Federated Averaging (FedAvg):** `FedAvg` is popularly used in many existing frameworks of FL. `FedAvg` enables local entities to compute more than one batch update on the local data and share the model weights instead of gradients. Besides, once all local entities begin from the same initialization process, the average of model weights is considered equivalent to averaging the gradients.

In general, FL provides a privacy-preserving paradigm for training the model, which allows participants to reserve their data locally [72]. Nevertheless, recent studies show that FL is unable to provide enough privacy protection [73, 74]. It is vulnerable to be attacked because attackers may infer participants' sensitive information from the global model. Besides, the communicating gradients may leak privacy as well. For instance, Zhu *et al.* [75] and Agarwal *et al.* [76] presented that gradients might reveal the sensitive information about local data. Zhao *et al.* [77] also confirmed that a malicious attack could obtain training data from gradients. Therefore, additional measures should be taken to prevent gradients or models from privacy leakage, which further motivates us to introduce LDP in FL in Chapter 3.

2.3.1 Existing Studies on Federated Learning for IoT

Recently, FL is explored extensively in IoT applications [78–82]. Lim *et al.* [78] surveyed FL applications in mobile edge networks comprehensively, including algorithms, applications, and potential research problems, etc. Besides, Lu *et al.* [80]

proposed an edge computing assisted collaborative machine learning framework for connected vehicles. Also, this proposed framework could reduce the training time while guaranteeing the accuracy of prediction. Moreover, Fantacci *et al.* [81] leveraged FL to protect the privacy of MEC, while Saputra *et al.* [82] applied FL to predict the energy demand for electrical vehicle networks. Zhao *et al.* [83] proposed a SecProbe mechanism to protect the privacy and quality of participants' data by leveraging exponential mechanism and functional mechanism of differential privacy. SecProbe guaranteed high accuracy as well as privacy protection. In addition, it prevented unreliable participants in collaborative learning. Lyu *et al.* [84] proposed a privacy-preserving deep learning framework, where a two-level protection mechanism, including *Random Projection* and *Differentially Private SGD*, was leveraged to protect the data privacy. Jiang *et al.* [85] designed a collaborative training method to protect features' privacy. In detail, the feature extraction was done locally in the devices such as smartphones while the classification was executed in the cloud service. Wang *et al.* [86, 87] proposed control algorithms to solve the problem of low resources in IoT devices while participating in FL.

Moreover, FL has attracted substantial attention recently [78, 88–91], and one of the most important issues in FL is privacy protection, which is explored in [79, 92–95]. Li *et al.* [92] considered the privacy issue during sharing model updates in FL. They proposed to leverage the sketch algorithms to build the sketching-based FL, which provided privacy guarantees while maintaining accuracy. Hao *et al.* [79] proposed a privacy-enhanced FL scheme to solve the privacy issue in FL. Their scheme helped to achieve efficient and privacy-preserving FL. Dolui *et al.* [93] applied FL paradigms to recommender systems and matrix factorization, which guaranteed the recommender systems' functionality and privacy. Nasr *et al.* [94] performed a comprehensive privacy analysis with white-box inference attacks. Wang *et al.* [95] proposed a framework incorporating a generative adversarial network with a multitask discriminator to solve the user-level privacy leakage in FL against attacks from a malicious server.

Furthermore, there have been many studies on FL and differential privacy such as [79, 84, 96–102]. For example, Truex *et al.* [97] utilized technologies like secure multiparty computation and centralized differential privacy to prevent inference over both the messages exchanged in the process of training the model. However, they did not analyze the impact of the privacy budget on the performance of

FL. Hu *et al.* [98] came up with a privacy-preserving FL approach for learning effective personalized models. They used the Gaussian mechanism, a centralized DP mechanism, to protect privacy. Hao *et al.* [79] proposed a differential enhanced FL scheme for the artificial industrial industry. Triastcyn *et al.* [100] employed the Bayesian differential privacy on FL. They used the centralized differential privacy mechanism to ensure the privacy of gradients, but we leverage a stronger privacy-preserving mechanism (LDP) to protect each vehicle's privacy. Additionally, DP can be applied to various FL algorithms such as `FedSGD` [71] and `FedAvg` [32]. `FedAvg` requires users to upload model parameters instead of gradients in `FedSGD`. The advantage of `FedAvg` is that it allows users to train the model for multiple rounds locally before submitting gradients. McMahan *et al.* [96] proposed to apply centralized DP to `FedAvg` and `FedSGD` algorithms.

In addition to DP, secure multiparty computation (SMPC) and homomorphic encryption (HE) are applied to FL for protecting the privacy of participants' data. SMPC ensures that different participants jointly collaborate while keeping inputs secured [103]. It ensured the secured data sharing among participants. The disadvantage of SMPC is its high communication and computation overhead, which slows down the efficiency of FL and increases the cost. Xu *et al.* [104] proposed an approach *HybridAlpha* for preserving privacy of the FL by combining differential privacy technique and SMPC. Their approach improved the efficiency of FL, but they did not propose new differential privacy mechanisms. Li *et al.* [105] proposed a SMPC based protocol for privacy-preserving feature selection to help build a more accurate FL model. Besides SMPC, HE is also adopted to secure FL. With HE, machine learning can conduct on the encrypted dataset without the accuracy loss. Since HE involves very high computational complexity, it is inefficient to use it in deep learning. Zhang *et al.* [106] customized a *BatchCrypt* system for cross-silo FL to reduce the encryption and communication overhead in HE. They developed new encoding and quantization schemes as well as a novel gradient clipping technique to replace the traditional encrypting method.

2.3.2 Existing Studies on Privacy-Preserving Crowdsourcing

Many studies focus on privacy-preserving crowdsourcing [107, 108], and leveraging the fog computing or edge computing to improve the performance as they have gained popularity [109–114]. For example, Wu *et al.* [107] proposed two generic models for quantifying mobile users' privacy and data utility in crowdsourced location-based services, respectively. He *et al.* [109] designed a privacy model for the crowdsourced bus service, which took advantage of the computational power of fog computing. However, their models apply only to the traditional crowdsourcing approach (i.e., customers transmit data to a centralized server) without considering the FL crowdsourcing tasks which leverage locally trained models. Zhao *et al.* [111] presented with a privacy-preserving mechanism to prevent the poisoning attack to the MEC. However, users need to offload data to the MEC server, which may leak privacy; instead, we propose that users retain their data locally.

Unlike the above studies, our proposed approach utilizes LDP noises to protect the privacy of the uploaded data. Our proposed LDP mechanisms provide stronger privacy protection using the LDP mechanism. In Chapter 3, we deploy LDP mechanisms to gradients in `FedSGD` algorithm.

2.4 Blockchain

In this section, concepts of the blockchain, Ethereum, smart contract, and IPFS are introduced.

Blockchain. The blockchain technology is popularly used in systems requiring high security and transparency, such as Bitcoin and Ethereum [115]. The blockchain can effectively solve the double-spending problem in Bitcoin transactions by using a peer-to-peer network. The solution is to hash transaction information in a chain of hash-based Proof-of-Work (PoW, used by Bitcoin), and then the consensus mechanism algorithm is used to confirm transactions and produce new blocks to the chain. Once the record is formed, it cannot be changed except redoing PoW.

Besides, the blockchain is constantly growing with appending 'completed' blocks. Blocks consisting of the most recent transactions are added to the chain in chronological order [116]. Each mining node can have a copy of the blockchain. The blockchain allows participants to track their transactions without the centralized control.

Ethereum. Ethereum is a blockchain platform that allows users to create decentralized, and end-to-end applications [117]. The miners in Ethereum use the Proof-of-Work consensus algorithm to complete verification and synchronization for transactions.

Smart Contract. Nick Szabo first proposes an automatic computerized transaction protocol to execute contract terms automatically, which later refers to as smart contract [118]. It intends to make a contract digitally and allows to maintain credible transactions without a third party. Blockchains' development, such as Ethereum and smart contracts, are stored in the blockchain as scripts. A blockchain with a Turing-complete programming language allows everyone to customize smart contract scripts for transactions [119]. Smart contracts are triggered when transactions are created or generated on the blockchain to finish specific tasks or services.

InterPlanetary File System (IPFS). The IPFS, which enables distributed devices with the computing capacity to connect with the same file system, is a peer-to-peer distributed file system. The implementation of the off-chain storage can use IPFS [49], and hashes of data locations instead of actual files can be stored on the blockchain. The hash can be used to locate the exact file across the system.

2.4.1 Existing Studies on Leveraging Blockchain for Privacy Protection

Blockchain is a fast-growing technology to provide security and privacy in a decentralized manner [42, 120–124]. Feng *et al.* [125] summarized prior studies about privacy protection in the blockchain system, including the methodology for identity and transaction privacy preservation. In the following, we will introduce more recent studies utilizing blockchain to provide privacy or security protection in identity, data, and transactions.

Leveraging Blockchains for Identity Privacy/Security Protection. A few studies have focused on leveraging the blockchain to guarantee privacy/security in access control management or identity protection. For example, Zyskind *et al.* [126] and Xia *et al.* [127] both used blockchain in access control management. Zyskind *et al.* [126] created a decentralized system for personal data management in order to address users' concerns about privacy when using third-party mobile platforms. Xia *et al.* [127] proposed a permissioned blockchain-based data-sharing framework to allow only verified users to access the cloud data. Lu *et al.* [128] developed a private and anonymous decentralized crowdsourcing system ZebraLancer, which overcame data leakage and identity breach in traditional decentralized crowdsourcing. The above studies focus on identity privacy because the blockchain is anonymous, whereas they do not consider the privacy protection for the database.

Leveraging Blockchains for Data Privacy/Security Protection. In addition to the identity privacy preservation, Hu *et al.* [129] utilized a smart contract instead of the central server and developed a privacy-preserving, at the same time, decentralized approach for computing encrypted data while ensuring the privacy of data to prevent from misbehaving of a malicious centralized server. Luongo *et al.* [130] used secure multi-party computation to design a privacy primitive named Keep, which enabled contracts to leverage private data without exposing the data to the public blockchain for protecting smart contracts on public blockchains. Alternatively, we use differential privacy standards to guarantee privacy. Moreover, blockchains are popular to be used for security protection of data sharing in IoT scenarios [121, 122, 127].

Leveraging Blockchains for Transaction Privacy/Security Protection. Moreover, some previous studies use blockchain to guarantee security and privacy in transactions. For example, Henry *et al.* [41] proposed that the blockchain should use mechanisms that piggyback on the overlay network, which was ready for announcing transactions to de-link users' network-level information instead of using an external service such as Tor to protect users' privacy. Gervais [131] proposed a quantitative framework to analyze the security of proof-of-work in blockchains, where the framework's inputs included security, consensus, and network parameters. Pérez-Solà and Herrera-Joancomartí [132] focused on protecting the privacy of bitcoin transactions. Sani *et al.* [133] proposed a new blockchain Xyreum with high-performance and scalability to secure transactions in the Industrial IoT.

Nevertheless, none of the existing work leverages the blockchain to manage differential privacy costs or reuse previous query results or noises. Given the above, in Chapter 4, we design an algorithm to reuse previous noisy responses if the same query is asked repeatedly. In particular, considering that different requests of the same query may have different privacy requirements, our algorithm can set the optimal reuse fraction of the old noisy response and add new noise to minimize the accumulated privacy costs. Furthermore, we design and implement a blockchain-based system for tracking and saving differential privacy costs. As a result, the dataset owner will have full knowledge about how the dataset has been used and be confident that no new privacy cost will be incurred for answering queries once the specified privacy budget is exhausted.

2.4.2 Existing Studies on Leveraging Blockchain for Differential-Privacy Costs Management

Yang *et al.* [43] utilized blockchain and differential privacy technologies to achieve security and privacy protection during data sharing. Compared with [43], we summarize the differences between our work and [43] as follows.

- Although Algorithm 1 of [43] claims to satisfy ϵ-*differential privacy*, it does not since the noisy output's domain (i.e., the set of all possible values) depends on the input. The explanation is as follows. In [43], for two neighboring datasets D and D', there exists a subset $\mathcal{Y}$ of outputs such that the probability $\mathbb{P}\left[\widetilde{Q}(D) \in \mathcal{Y}\right] > 0$ but $\mathbb{P}\left[\widetilde{Q}(D') \in \mathcal{Y}\right] = 0$, where $\mathbb{P}$ denotes probability and $\widetilde{Q}(D)$ represents the noisy query response for query Q on dataset D. This means $\frac{\mathbb{P}\left[\widetilde{Q}(D)\in\mathcal{Y}\right]}{\mathbb{P}\left[\widetilde{Q}(D')\in\mathcal{Y}\right]} = \infty > e^{\epsilon}$, which violates ϵ-*differential privacy* for any $\epsilon < \infty$.

- [43] does not discuss how to choose the small additional privacy parameter in its Algorithm 1.

- In [43], when a query is asked for the first time, the `Laplace` mechanism of [7] for ϵ-*differential privacy* is used to add Laplace noise to the true query result. Afterward, [43] adds the new Laplacian noise on the previous noisy output, which makes the new noisy response no longer follow Laplace distribution

since the sum of independent Laplace random variables does not follow a Laplace distribution. Hence, the analysis in [43] is not effective.

The Gaussian noise is used to achieve (ϵ, δ)-differential privacy. The advantage of the Gaussian noise over Laplace noise lies in the easier privacy analysis for the composition of different privacy-preserving algorithms since the sum of independent Gaussian random variables follows the Gaussian distribution. In contrast, the sum of independent Laplace random variables does not obey the Laplace distribution.

2.4.3 Existing Studies on Leveraging Blockchain for Federated Learning

Blockchain and FL techniques have been widely used in training a neural network with distributed data [33, 83, 134–141]. For example, Weng *et al.* [140] proposed a system called DeepChain for collaborative learning. But they did not offload the training task to the edge server, and they did not propose using DP to guarantee the privacy of model parameters. Awan *et al.* [139] proposed a blockchain-based privacy-preserving FL framework, which secured the model update using blockchain's immutability and decentralized trust properties. Li *et al.* [142] designed a decentralized framework based on blockchain for crowdsourcing tasks, which enabled them to do crowdsourcing tasks without a centralized server. Lu *et al.* [135] proposed to leverage blockchain, FL, and DP for data sharing. However, they directly added the DP noise to the original data instead of the gradients, seriously affecting accuracy. Lyu *et al.* [141] made the first-ever investigation on the federated fairness in a blockchain-assisted decentralized deep learning framework, and they enforced fairness by designing a local credibility mutual evaluation mechanism. They also developed a scheme for encryption to ensure privacy and accuracy.

2.5 Summary

In conclusion, in this chapter, studies related to privacy-preserving data analysis are surveyed. In particular, we focus on the literature that leverages DP, LDP,

FL, and blockchain related technologies. Besides, we point out the disadvantages of the existing solutions and highlight the novelty of our proposed approaches.

Chapter 3

Local Differential Privacy based Federated Learning for IoT[1]

In this chapter, we study novel LDP mechanisms for estimating the mean value over multiple-dimensional numeric attributes and empirical risk minimization tasks. The main contributions of this chapter are summarized as follows:

- Using an LDP based federated stochastic gradient descent algorithm (`LDP-FedSGD`) for FL in the Internet of Vehicles (IoV) as a motivating context, we present novel LDP mechanisms for numeric data with a continuous domain. Among our proposed mechanisms, `Three-Outputs` and `PM-SUB` outperform existing mechanisms for a wide range of privacy parameter ϵ, as shown in the theoretical results in Table 3.1 and confirmed by experiments. In terms of comparing `Three-Outputs` and the suboptimal piecewise mechanism (`PM-SUB`), we have: `Three-Outputs`, whose output has three possibilities, achieves a smaller worst-case variance when ϵ is small. In contrast, `PM-SUB`, whose output can take infinite possibilities of an interval, has a smaller worst-case variance for large ϵ. `PM-SUB` is a slightly suboptimal version of the optimal piecewise mechanism `PM-OPT` aiming to simplify the expressions. We further combine `Three-Outputs` and `PM-SUB` to obtain a hybrid mechanism `HM-TP`, which obtains an even smaller worst-case variance.

[1]The work in this chapter has been published as Yang Zhao, Jun Zhao, Mengmeng Yang, Teng Wang, Ning Wang, Lingjuan Lyu, Dusit Niyato, and Kwok-Yan Lam. "Local Differential Privacy Based Federated Learning for Internet of Things", in *IEEE Internet of Things Journal, DOI: 10.1109/JIOT.2020.3037194, 2020.*

TABLE 3.1: A comparison of the worst-case variances of our and existing ϵ-LDP mechanisms on a single numeric attribute with a domain $[-1, 1]$. `Three-Outputs` and `PM-SUB` are our main LDP mechanisms proposed in this chapter. The results in this table show the advantages of our mechanisms over existing mechanisms for a wide range of privacy parameter ϵ. For an LDP mechanism $\mathcal{M}$, its worst-case variance is denoted by $V_{\mathcal{M}}$. We obtain this table based on results of [18].

Range of ϵ	Comparison of mechanisms
$0 < \epsilon < \ln 2 \approx 0.69$	$V_{\texttt{Duchi}} = V_{\texttt{Three-Outputs}} < V_{\texttt{PM-SUB}} < V_{\texttt{PM}}$
$\ln 2 < \epsilon < 1.19$	$V_{\texttt{Three-Outputs}} < V_{\texttt{Duchi}} < V_{\texttt{PM-SUB}} < V_{\texttt{PM}}$
$1.19 < \epsilon < 1.29$	$V_{\texttt{Three-Outputs}} < V_{\texttt{PM-SUB}} < V_{\texttt{Duchi}} < V_{\texttt{PM}}$
$1.29 < \epsilon < 2.56$	$V_{\texttt{Three-Outputs}} < V_{\texttt{PM-SUB}} < V_{\texttt{PM}} < V_{\texttt{Duchi}}$
$2.56 < \epsilon < 3.27$	$V_{\texttt{PM-SUB}} < V_{\texttt{Three-Outputs}} < V_{\texttt{PM}} < V_{\texttt{Duchi}}$
$\epsilon > 3.27$	$V_{\texttt{PM-SUB}} < V_{\texttt{PM}} < V_{\texttt{Three-Outputs}} < V_{\texttt{Duchi}}$

- We discretize the continuous output ranges of our proposed mechanisms `PM-SUB` and `PM-OPT`. Through the discretization post-processing, we enable vehicles to use our proposed mechanisms. In Section 3.5, we confirm with our experiments that the discretization post-processing algorithm maintains a small worst-case variance while reducing the communication cost.

- Experimental evaluation of our proposed mechanisms on real-world and synthetic datasets demonstrates that our proposed mechanisms achieve higher accuracy in estimating the mean frequency of the data and performing empirical risk minimization tasks than existing approaches.

In the following, in Section 3.1, we illustrate the system model and LDP based `FedSGD` algorithm. Section 3.2 presents the problem formation. Section 3.3 proposes novel solutions for the single numerical data estimation. Section 3.4 illustrates proposed mechanisms used for multidimensional numerical data estimation. Section 3.5 demonstrates our experimental results. Section 3.6 summarizes this chapter. Notations used in the rest of this chapter are summarized in Table 3.2.

TABLE 3.2: Summary of notations.

ϵ	privacy parameter
$\mathbb{P}[\cdot]$	probability
$P_{y \leftarrow x}$	shortage of $\mathbb{P}[Y = y \mid X = x]$
$\mathbb{F}[\cdot]$	probability density function
$\nabla \ell$	gradient
η	learning rate
G	a group of vehicles
$\mathcal{M}$	local differential privacy mechanism $\mathcal{M}$
$\mathbb{E}$	expectation
Var	variance
$\mathbb{Y}$	range of the noisy output
x	input value
L	left boundary of the center piece of `PM-OPT`'s probability density function
R	right boundary of the center piece of `PM-OPT`'s probability density function

3.1 System Model and Local Differential Privacy based FedSGD Algorithm

3.1.1 System Model

We consider a scenario where a number of vehicles are connected with a cloud server as shown in Fig. 3.1. Each vehicle is responsible for continuously performing training and inference based on data collected locally and the model initiated by the cloud server. The local training dataset does not need to be uploaded to the cloud server. After finishing predefined epochs locally, the cloud server calculates the average of uploaded gradients from vehicles and updates the global model with the averaged model parameters. The FL aggregator is honest-but-curious or semi-honest, which obeys the FL protocol and tries to learn additional information using received data [91, 97]. With the injected LDP noise, servers or attackers cannot retrieve users' information by reversing their uploaded gradients [38, 39]. Thus, there is a need to deploy LDP mechanisms to the FL to develop a communication-efficient LDP-based FL algorithm.

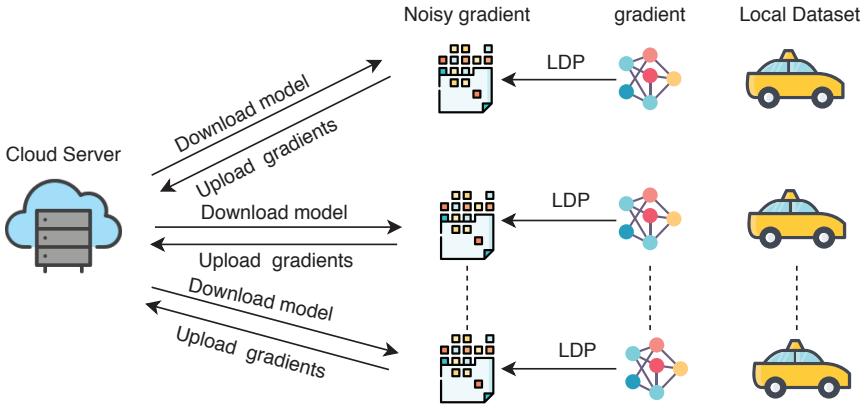


FIGURE 3.1: System Design.

3.1.2 Federated learning with LDP: LDP-FedSGD

In addition, we propose an `LDP-FedSGD` mechanism for our proposed system. Details of `LDP-FedSGD` are given in Algorithm 1. Unlike the `FedAvg` algorithm, in the `FedSGD` algorithm, each client (i.e., vehicle) uploads the updated gradient $\nabla\ell$ instead of model parameters to the central aggregator (i.e., cloud server) [71]. However, compared with the standard `FedSGD` [71], we add our proposed LDP mechanism proposed in Section 3.4 to prevent the privacy leakage of gradients. Each vehicle takes one step of gradient descent on the current model locally with its local data, and then it perturbs the true gradient with Algorithm 6. The server aggregates and averages the updated gradients from vehicles and then updates the model. To reduce the communication rounds, we separate vehicles into groups so that the cloud server updates the model after gathering gradient updates from vehicles in a group.

3.1.3 Comparing LDP-FedSGD with other privacy-preserving federated learning paradigms

The `LDP-FedSGD` algorithm incorporates LDP noises into gradients of FL. In addition to `LDP-FedSGD`, one may be interested in other ways of using DP in FL. To explain them, we categorize combinations of DP and FL (or distributed computations in general) by considering the place of perturbation (distributed/centralized perturbation) and privacy granularity (user-level/record-level privacy protection) [143]:

Algorithm 1: Local Differential Privacy based FedSGD (`LDP-FedSGD`) Algorithm.

Server executes:
Server initializes the parameter as θ_0;
for *t from* 1 *to maximal iteration number* **do**
  Server sends θ_{t-1} to vehicles in group G_t;
  for *each vehicle i in Group* G_t **do**
    VehicleUpdate(i, $\nabla\ell$):
  Server computes the average of the noisy gradient of group G_t and updates the parameter from θ_{t-1} to θ_t: $\theta_t \leftarrow \theta_{t-1} - \eta_t \cdot \frac{1}{|G_t|} \sum_{i \in G_t} \mathcal{M}(\nabla\ell(\theta_{t-1}; x_i))$, where η_t is the learning rate and x_i is vehicle i's data;
  if θ_t *and* θ_{t-1} *are close enough or these remains no vehicle which has not participated in the computation* **then**
    break;
  $t \to t+1$;

VehicleUpdate (i, $\nabla\ell$):
Compute the (true) gradient $\nabla\ell(\theta_{t-1}; x_i)$;
Use local differential privacy-compliant algorithm $\mathcal{M}$ to compute the noisy gradient $\mathcal{M}(\nabla\ell(\theta_{t-1}; x_i))$;

- **Distributed/Centralized perturbation.** Note that differential privacy is achieved by introducing perturbation. Distributed perturbation considers an honest-but-curious aggregator, while the centralized perturbation needs a trusted aggregator. Both perturbation methods defend against external inference attacks after model publishing.

- **User-level/Record-level privacy protection.** In general, a differentially private algorithm ensures that the probability distributions of the outputs on two neighboring datasets do not differ much. The distinction between user-level and record-level privacy protection lies in how neighboring datasets are defined. We define that two datasets are user-neighboring if one dataset can be formed from the other dataset by adding or removing all one user's records arbitrarily. If one dataset is achieved by changing a *single* record of the other dataset from one user, they are called record-neighboring.

Based on the above, four paradigms are obtained as follows: 1) **ULDP** (user-level privacy protection with distributed perturbation), 2) **RLDP** (record-level privacy protection with distributed perturbation), 3) **RLCP** (record-level privacy protection with centralized perturbation), and 4) **ULCP** (user-level privacy protection with centralized perturbation). The details can be found in the related work [143].

In the case of distributed perturbation, when all users set the same privacy parameter ϵ, we refer to ULDP and RLDP above as ϵ-ULDP and ϵ-RLDP, respectively. Table 3.3 presents a comparison of ϵ-ULDP, ϵ-RLDP, ϵ-RLCP, and ϵ-ULCP. In this chapter, each user applies ϵ-LDP, so our framework is under ULDP. The reasons that we consider ULDP instead of RLDP, RLCP, and ULCP are as follows.

- We do not consider RLDP, which implements perturbation at each user via standard differential privacy. We aim to achieve user-level privacy protection instead of the weaker record-level privacy protection (a vehicle user in our IoV applications may have multiple records). The motivation is that often much data from a vehicle may be about the vehicle's regular driver, and it often makes more sense to protect all data about the regular driver instead of just protecting every single record. A similar argument has been recently stated in [96], which incorporates user-level differential privacy into the training process of FL for language modeling. Specifically, [96] considers user-level privacy to protect the privacy of all typed words of a user and explains that such privacy protection is more reasonable than protecting individual words as in the case of record-level privacy. In addition, although we can compute the level of user-level privacy from record-level privacy via the group privacy property of differential privacy (see Theorem 2.2 of [8]), this may significantly increase the privacy parameter and hence weaken the privacy protection if a user has many records (note that a more significant privacy parameter ϵ in ϵ-DP means weaker privacy protection). More specifically, for a user with m records, according to the group privacy property [8], the privacy protection strength for the user under ϵ-record-level privacy is just as that under $m\epsilon$-user-level privacy (i.e., for a user with m records, ϵ-RLDP ensures $m\epsilon$-ULDP; ϵ-RLCP ensures $m\epsilon$-ULCP).

- We do not investigate RLCP and ULCP since this chapter considers an honest-but-curious aggregator instead of a trusted aggregator. The aggregator is not entirely trusted, so the perturbation is implemented at each user (i.e., vehicle in IoV).

TABLE 3.3: We compare different privacy notions in this table. In this chapter, we focus on ϵ-LDP which achieves user-level privacy protection with distributed perturbation (ULDP). We do not consider record-level privacy protection with distributed perturbation (RLDP) which implements perturbation at each user via standard differential privacy, since we aim to achieve user-level privacy protection instead of the weaker record-level privacy protection (a vehicle is a user in our IoV applications and may have multiple records). We also do not investigate record/user-level privacy protection with centralized perturbation (RLCP/ULCP) since this chapter considers a honest-but-curious aggregator instead of a trusted aggregator.

privacy granularity and place of perturbation	privacy property	adversary model
ϵ-LDP (defined for **distributed** perturbation)	ϵ-ULDP	defend against a honest-but-curious aggregator & external attacks after model publishing
ϵ-DP with **distributed** perturbation	ϵ-RLDP	
ϵ-DP with **centralized** perturbation	ϵ-RLCP	trusted aggregator; defend against external attacks after model publishing
user-level privacy with **centralized** perturbation	ϵ-ULCP	

3.2 Problem Formation

Let x be a user's true value, and Y be the perturbed value. Under the perturbation mechanism $\mathcal{M}$, we use $\mathbb{E}_{\mathcal{M}}[Y|x]$ to represent the expectation of the randomized output Y given input x. $\mathrm{Var}_{\mathcal{M}}[Y|x]$ is the variance of output Y given input x. $\mathrm{MaxVar}(\mathcal{M})$ denotes the worst-case $\mathrm{Var}_{\mathcal{M}}[Y|x]$. Our target is to find a privatization mechanism $\mathcal{M}$ that minimizes $\mathrm{MaxVar}(\mathcal{M})$ by solving the following constraint minimization problem:

$$\begin{aligned} &\min_{\mathcal{M}} \mathrm{MaxVar}(\mathcal{M}), \\ &\text{s.t. Eq. (2.4)}, \\ &\quad \mathbb{E}_{\mathcal{M}}[Y|x] = x, \text{ and} \\ &\quad \mathbb{P}_{\mathcal{M}}[Y \in \mathbb{Y}|x] = 1. \end{aligned}$$

The second constraint illustrates that our estimator is unbiased, and the third constraint shows the proper distribution where $\mathbb{Y}$ is the range of randomized function $\mathcal{M}$. In the following sections, if $\mathcal{M}$ is clear from the context, we omit the subscript $\mathcal{M}$ for simplicity.

3.3 Mechanisms for Estimation of A Single Numeric Attribute

We propose four LDP mechanisms: `Three-Outputs`, `PM-OPT`, `PM-SUB`, and `HM-TP` in order to solve the problem in Section 3.2. Fig. 3.2 compares the worst-case noise variances of existing mechanisms and our proposed mechanisms. `Three-Outputs` has three discrete output possibilities, which incurs little communication cost because two bits are enough to encode three different outputs. Moreover, it achieves a small worst-case noise variance in the high privacy regime (small privacy budget ϵ). However, to maintain a low worst-case noise variance in the low privacy regime (large privacy budget ϵ), we propose `PM-OPT` and `PM-SUB`. Both of them achieve higher accuracies than `Three-Outputs` and other existing solutions when the privacy budget ϵ is large. Additionally, we discretize their continuous ranges of output for vehicles to encode using a post-processing discretization algorithm. In the following sections, we will explain our proposed four mechanisms and the post-processing discretization algorithm in detail.

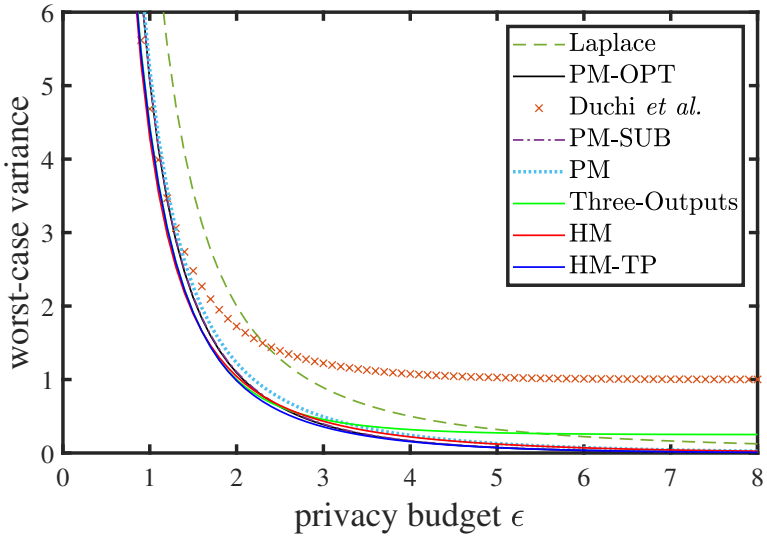


FIGURE 3.2: The worst-case of different mechanisms' noise variance for one-dimensional numeric data w.r.t. the privacy budget ϵ.

3.3.1 `Three-Outputs` Mechanism

We propose a mechanism with three output possibilities named as `Three-Outputs` which is illustrated in Algorithm 2. `Three-Outputs` ensures low communication cost while achieving a smaller worst-case noise variance than existing solutions in the high privacy regime (small privacy budget ϵ). `Duchi` *et al.*'s [63] solution contains two output possibilities, and it outperforms other approaches when the privacy budget is small. However, when ϵ increases, two outputs are not always optimal [144]. By outputting three values instead of two, `Three-Outputs` improves the performance as the privacy budget increases, which is shown in Fig. 3.2. When the privacy budget is small, `Three-Outputs`' worst-case variance is equivalent to that of `Duchi` *et al.*'s [63] solution.

For notional simplicity, given a mechanism $\mathcal{M}$, we often write $\mathbb{P}_{\mathcal{M}}[Y = y \mid X = x]$ as $P_{y \leftarrow x}(\mathcal{M})$ below. We also sometimes omit $\mathcal{M}$ to obtain $\mathbb{P}[Y = y \mid X = x]$ and $P_{y \leftarrow x}$.

Algorithm 2: `Three-Outputs` Mechanism for One-Dimensional Numeric Data.

Input: tuple $x \in [-1, 1]$ and privacy parameter ϵ.
Output: tuple $Y \in \{-C, 0, C\}$.
Sampling a random variable u with the probability distribution as follows:

$$\begin{aligned} \mathbb{P}[u = -1] &= P_{-C \leftarrow x}, \\ \mathbb{P}[u = 0] &= P_{0 \leftarrow x}, \text{ and} \\ \mathbb{P}[u = 1] &= P_{C \leftarrow x}, \end{aligned}$$

where $P_{-C \leftarrow x}$, $P_{0 \leftarrow x}$ and $P_{C \leftarrow x}$ are given in Eq. (3.1), Eq. (3.2) and Eq. (3.3).
if $u = -1$ **then**
  $Y = -C$;
else if $u = 0$ **then**
  $Y = 0$;
else
  $Y = C$;
return Y;

When a tuple $x \in [-1, 1]$, `Three-Outputs` outputs a perturbed value Y which is equivalent to $-C$, 0 or C with C defined as $C = \frac{e^{\epsilon}(e^{\epsilon}+1)}{(e^{\epsilon}-1)(e^{\epsilon}-P_{0 \leftarrow 0})}$ and probabilities

defined by

$$P_{-C\leftarrow x}=\begin{cases}\frac{1-P_{0\leftarrow 0}}{2}+\left(\frac{1-P_{0\leftarrow 0}}{2}-\frac{e^{\epsilon}-P_{0\leftarrow 0}}{e^{\epsilon}(e^{\epsilon}+1)}\right)x, \text{if } 0\leq x\leq 1,\\ \frac{1-P_{0\leftarrow 0}}{2}+\left(\frac{e^{\epsilon}-P_{0\leftarrow 0}}{e^{\epsilon}+1}-\frac{1-P_{0\leftarrow 0}}{2}\right)x, \text{if } -1\leq x\leq 0,\end{cases} \tag{3.1}$$

$$P_{C\leftarrow x}=\begin{cases}\frac{1-P_{0\leftarrow 0}}{2}+\left(\frac{e^{\epsilon}-P_{0\leftarrow 0}}{e^{\epsilon}+1}-\frac{1-P_{0\leftarrow 0}}{2}\right)x, \text{if } 0\leq x\leq 1,\\ \frac{1-P_{0\leftarrow 0}}{2}+\left(\frac{1-P_{0\leftarrow 0}}{2}-\frac{e^{\epsilon}-P_{0\leftarrow 0}}{e^{\epsilon}(e^{\epsilon}+1)}\right)x, \text{if } -1\leq x\leq 0,\end{cases} \tag{3.2}$$

$$\text{and } P_{0\leftarrow x}=P_{0\leftarrow 0}+(\frac{P_{0\leftarrow 0}}{e^{\epsilon}}-P_{0\leftarrow 0})x, \text{ if } -1\leq x\leq 1, \tag{3.3}$$

where $P_{0\leftarrow 0}$ is defined by

$$P_{0\leftarrow 0}:=\begin{cases}0, \quad \text{if } \epsilon<\ln 2,\\ -\frac{1}{6}(-e^{2\epsilon}-4e^{\epsilon}-5\\ +2\sqrt{\Delta_0}\cos(\frac{\pi}{3}+\frac{1}{3}\arccos(-\frac{\Delta_1}{2\Delta_0^{\frac{3}{2}}}))), \text{if } \ln 2\leq\epsilon\leq\epsilon',\\ \frac{e^{\epsilon}}{e^{\epsilon}+2}, \text{ if } \epsilon>\epsilon',\end{cases} \tag{3.4}$$

in which

$$\Delta_0:=e^{4\epsilon}+14e^{3\epsilon}+50e^{2\epsilon}-2e^{\epsilon}+25, \tag{3.5}$$

$$\Delta_1:=-2e^{6\epsilon}-42e^{5\epsilon}-270e^{4\epsilon}-404e^{3\epsilon}-918e^{2\epsilon}+30e^{\epsilon}-250, \tag{3.6}$$

$$\text{and } \epsilon':=\ln\left(\frac{3+\sqrt{65}}{2}\right)\approx\ln 5.53. \tag{3.7}$$

Next, we will show how we derive the above probabilities. For a mechanism which uses $x\in[-1,1]$ as the input and only three possibilities $-C,0,C$ for the

output value, it satisfies

$$\begin{cases} \epsilon\text{-LDP: } \dfrac{P_{C\leftarrow x}}{P_{C\leftarrow x'}}, \dfrac{P_{0\leftarrow x}}{P_{0\leftarrow x'}}, \dfrac{P_{-C\leftarrow x}}{P_{-C\leftarrow x'}} \in [e^{-\epsilon}, e^{\epsilon}], & \text{(3.8a)} \\ \text{unbiased estimation:} \\ \quad C \cdot P_{C\leftarrow x} + 0 \cdot P_{0\leftarrow x} + (-C) \cdot P_{-C\leftarrow x} = x, & \text{(3.8b)} \\ \text{proper distribution:} \\ \quad P_{y\leftarrow x} \geq 0 \text{ and } P_{C\leftarrow x} + P_{0\leftarrow x} + P_{-C\leftarrow x} = 1. & \text{(3.8c)} \end{cases}$$

To calculate values of $P_{C\leftarrow x}$, $P_{0\leftarrow x}$ and $P_{-C\leftarrow x}$, we use Lemma 1 below to convert a mechanism $\mathcal{M}_1$ satisfying the requirements in (3.8a) (3.8b) (3.8c) to a symmetric mechanism $\mathcal{M}_2$. Then, we use Lemma 2 below to transform the symmetric mechanism further to $\mathcal{M}_3$ whose worst-case noise variance is smaller than $\mathcal{M}_2$'s. Next, we use $P_{0\leftarrow 1}$ to represent other probabilities, and then we prove that we get the minimum variance when $P_{0\leftarrow 0} = e^{\epsilon} P_{0\leftarrow 1}$ using Lemma 3. Finally, Lemma 4 and Lemma 5 are used to obtain values for $P_{0\leftarrow 0}$ and the worst-case noise variance of `Three-Outputs`, respectively. Thus, we can obtain values of $P_{C\leftarrow x}$, $P_{0\leftarrow x}$ and $P_{-C\leftarrow x}$ using $P_{0\leftarrow 0}$. In the following, we will illustrate above processes in detail.

By symmetry, for any $x \in [-1, 1]$, we enforce

$$\begin{cases} P_{C\leftarrow x} = P_{-C\leftarrow -x}, & \text{(3.9a)} \\ P_{0\leftarrow x} = P_{0\leftarrow -x}, & \text{(3.9b)} \end{cases}$$

where Eq. (3.9b) can be derived from Eq. (3.9a). The formal justification of Eq. (3.9a) and Eq. (3.9b) is given by Lemma 1 below. Since the input domain $[-1, 1]$ is symmetric, we can transform any mechanism satisfying requirements in (3.8a) (3.8b) (3.8c) to a symmetric mechanism while guaranteeing the worst-case noise variance will not increase in Lemma 1. Thus, we can derive probabilities when $x \in [-1, 0]$ using probabilities when $x \in [0, 1]$ based on the symmetry.

Lemma 1. For a mechanism $\mathcal{M}_1$ satisfying the requirements in (3.8a) (3.8b) (3.8c), the following symmetrization process to obtain a mechanism $\mathcal{M}_2$ will not increase (i.e., will reduce or not change) the worst-case noise variance, while mechanism $\mathcal{M}_2$ still satisfies the requirements in (3.8a) (3.8b) (3.8c). Symmetrization: For

$x \in [-1, 1]$,

$$P_{C \leftarrow x}(\mathcal{M}_2) = P_{-C \leftarrow -x}(\mathcal{M}_2) = \frac{P_{C \leftarrow x}(\mathcal{M}_1) + P_{-C \leftarrow -x}(\mathcal{M}_1)}{2}, \tag{3.10}$$

$$P_{0 \leftarrow x}(\mathcal{M}_2) = P_{0 \leftarrow -x}(\mathcal{M}_2) = \frac{P_{0 \leftarrow x}(\mathcal{M}_1) + P_{0 \leftarrow -x}(\mathcal{M}_1)}{2}. \tag{3.11}$$

Proof. The proof details are given in Appendix A.1. □

Based on Lemma 1, we define a symmetric mechanism as follows.

Symmetric Mechanism. A mechanism under requirements of (3.8a) (3.8b) (3.8c) is called a symmetric mechanism if it satisfies Eq. (3.9a) and Eq. (3.9b). In the following, we only consider the symmetric mechanism $\mathcal{M}_2$.

Now, we design probabilities for the symmetric mechanism $\mathcal{M}_2$. As $\mathcal{M}_2$ satisfies the unbiased estimation which is a linear relationship, we set probabilities as piecewise linear functions of x as follows:

Case 1: For $x \in [0, 1]$,

$$P_{C \leftarrow x} = P_{C \leftarrow 0} + (P_{C \leftarrow 1} - P_{C \leftarrow 0})x, \tag{3.12}$$

$$P_{-C \leftarrow x} = P_{-C \leftarrow 0} - (P_{-C \leftarrow 0} - P_{-C \leftarrow 1})x, \tag{3.13}$$

$$\begin{aligned} P_{0 \leftarrow x} &= 1 - P_{-C \leftarrow 0} - P_{C \leftarrow 0} \\ &\quad + (P_{-C \leftarrow 0} + P_{C \leftarrow 0} - P_{-C \leftarrow 1} - P_{C \leftarrow 1})x. \end{aligned} \tag{3.14}$$

Case 2: For $x \in [-1, 0]$,

$$P_{C \leftarrow x} = P_{C \leftarrow 0} + (P_{C \leftarrow 0} - P_{C \leftarrow -1})x, \tag{3.15}$$

$$P_{-C \leftarrow x} = P_{-C \leftarrow 0} - (P_{-C \leftarrow -1} - P_{-C \leftarrow 0})x, \tag{3.16}$$

$$\begin{aligned} P_{0 \leftarrow x} &= 1 - P_{-C \leftarrow 0} - P_{C \leftarrow 0} \\ &\quad + (P_{-C \leftarrow -1} - P_{-C \leftarrow 0} - P_{C \leftarrow 0} + P_{C \leftarrow -1})x. \end{aligned} \tag{3.17}$$

Then, we may assign values to our designed probabilities above. We find that if a symmetric mechanism satisfies Eq. (3.18a) and Eq. (3.18b), it obtains a smaller

worst-case noise variance. From Lemma 2 below, we enforce

$$\begin{cases} P_{C\leftarrow 1} = e^{\epsilon} P_{C\leftarrow -1}, & \text{(3.18a)} \\ P_{-C\leftarrow -1} = e^{\epsilon} P_{-C\leftarrow 1}. & \text{(3.18b)} \end{cases}$$

Hence, given a symmetric mechanism $\mathcal{M}_2$ satisfying Inequality (3.19), we can transform it to a new symmetric mechanism $\mathcal{M}_3$ which satisfies Eq. (3.18a) and Eq. (3.18b) through processes of Eq. (3.20), Eq. (3.21) and Eq. (3.22) until $P_{C\leftarrow -1} = e^{\epsilon} P_{-C\leftarrow 1}$. After transformation, the new mechanism $\mathcal{M}_3$ achieves a smaller worst-case noise variance than mechanism $\mathcal{M}_2$. Therefore, we use the new symmetric mechanism $\mathcal{M}_3$ to replace $\mathcal{M}_2$ in the future's discussion. Details of transformation are in the Lemma 2.

Lemma 2. For a symmetric mechanism $\mathcal{M}_2$, if

$$P_{C\leftarrow 1}(\mathcal{M}_2) < e^{\epsilon} P_{C\leftarrow -1}(\mathcal{M}_2), \tag{3.19}$$

we set a symmetric mechanism $\mathcal{M}_3$ as follows: For $x \in [-1, 1]$,

$$\begin{aligned} P_{C\leftarrow x}(\mathcal{M}_3) &= P_{-C\leftarrow -x}(\mathcal{M}_3) \\ &= P_{C\leftarrow x}(\mathcal{M}_2) - \frac{e^{\epsilon} P_{C\leftarrow -1}(\mathcal{M}_2) - P_{C\leftarrow 1}(\mathcal{M}_2)}{e^{\epsilon} - 1}, \end{aligned} \tag{3.20}$$

$$\begin{aligned} P_{-C\leftarrow x}(\mathcal{M}_3) &= P_{C\leftarrow -x}(\mathcal{M}_3) \\ &= P_{-C\leftarrow x}(\mathcal{M}_2) - \frac{e^{\epsilon} P_{-C\leftarrow 1}(\mathcal{M}_2) - P_{-C\leftarrow -1}(\mathcal{M}_2)}{e^{\epsilon} - 1}, \end{aligned} \tag{3.21}$$

$$\begin{aligned} \text{and } P_{0\leftarrow x}(\mathcal{M}_3) &= 1 - P_{C\leftarrow x}(\mathcal{M}_3) - P_{-C\leftarrow x}(\mathcal{M}_3) \\ &= P_{0\leftarrow x}(\mathcal{M}_2) + \frac{2(e^{\epsilon} P_{C\leftarrow -1}(\mathcal{M}_2) - P_{C\leftarrow 1}(\mathcal{M}_2))}{e^{\epsilon} - 1}. \end{aligned} \tag{3.22}$$

Moreover, the mechanism $\mathcal{M}_3$ has a worst-case noise variance smaller than that of $\mathcal{M}_2$, while $\mathcal{M}_3$ still satisfies the requirements in (3.8a) (3.8b) (3.8c).

Proof. The proof details are given in Appendix A.2. □

We have proved that the symmetric mechanism $\mathcal{M}_3$ has a smaller worst-case noise variance than that of mechanism $\mathcal{M}_2$ in Lemma 2. Then, we use mechanism $\mathcal{M}_3$

to obtain the relation between $P_{0\leftarrow 1}$ and $P_{0\leftarrow 0}$ to find the minimum variance. From Lemma 3 below, we enforce

$$P_{0\leftarrow 0} = e^{\epsilon} P_{0\leftarrow 1}. \tag{3.23}$$

Then, we use the following Lemma 3 to obtain the relation between $P_{0\leftarrow 1}$ and $P_{0\leftarrow 0}$, so that we can obtain $P_{C\leftarrow x}$, $P_{0\leftarrow x}$ and $P_{-C\leftarrow x}$ using $P_{0\leftarrow 0}$.

Lemma 3. Given $P_{0\leftarrow 0}$, the variance of the output given input x is a strictly increasing function of $P_{0\leftarrow 1}$ and hence is minimized when $P_{0\leftarrow 1} = \frac{P_{0\leftarrow 0}}{e^{\epsilon}}$.

Proof. The proof details are given in Appendix A.3. □

Lemma 3 shows that we get the minimum variance when $P_{0\leftarrow 1} = \frac{P_{0\leftarrow 0}}{e^{\epsilon}}$. Hence, we replace $e^{\epsilon} P_{0\leftarrow 1}$ with $P_{0\leftarrow 0}$. Then, the variance is equivalent to

$$\mathrm{Var}[Y|X=x] = \left(\frac{e^{\epsilon}+1}{(e^{\epsilon}-1)(1-\frac{P_{0\leftarrow 0}}{e^{\epsilon}})}\right)^2 \left(1 - P_{0\leftarrow 0} + (P_{0\leftarrow 0} - \frac{P_{0\leftarrow 0}}{e^{\epsilon}})|x|\right) - x^2. \tag{3.24}$$

Complete details for obtaining Eq. (3.24) are in Appendix A.3.

Next, we use Lemma 4 to obtain the optimal $P_{0\leftarrow 0}$ in `Three-Outputs` to achieve the minimum worst-case variance as follows:

Lemma 4. The optimal $P_{0\leftarrow 0}$ to minimize the $\max_{x\in[-1,1]} \mathrm{Var}[Y|x]$ is defined by Eq. (3.4).

Proof. The proof details are given in Appendix A.4. □

Remark 3.1. Fig. 3.3 displays how $P_{0\leftarrow 0}$ changes with ϵ in Eq. (3.4). When the privacy budget ϵ is small, $P_{0\leftarrow 0} = 0$. Thus, `Three-Outputs` is equivalent to `Duchi` *et al.*'s [63] solution when $P_{0\leftarrow 0} = 0$. However, as the privacy budget ϵ increases, $P_{0\leftarrow 0}$ increases, which means that the probability of outputting true value increases.

By summarizing above, we obtain $P_{-C\leftarrow x}$, $P_{C\leftarrow x}$ and $P_{0\leftarrow x}$ from Eq. (3.1), Eq. (3.2) and Eq. (3.3) using $P_{0\leftarrow 0}$.

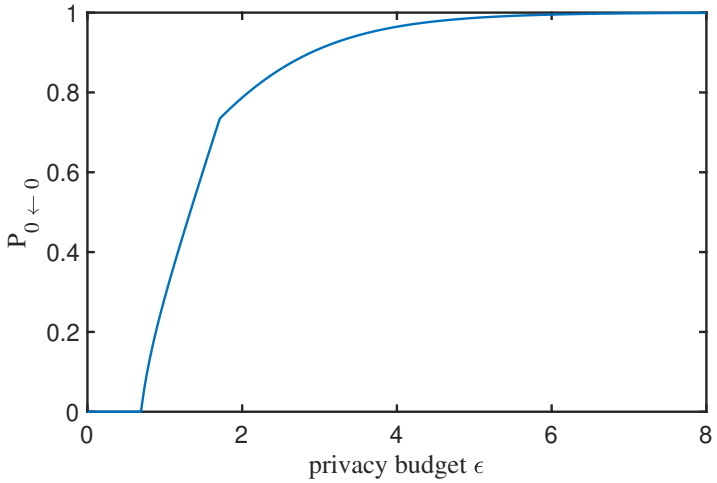


FIGURE 3.3: Optimal $P_{0\leftarrow 0}$ if the privacy budget $\epsilon \in [0, 8]$.

Then, we can calculate the optimal $P_{0\leftarrow 0}$ to obtain the minimum worst-case noise variance of `Three-Outputs` as follows:

Lemma 5. The minimum worst-case noise variance of `Three-Outputs` is obtained when $P_{0\leftarrow 0}$ satisfies Eq. (3.4).

Proof. The proof details are given in Appendix A.5. □

A clarification about `Three-Outputs` versus `Four-Outputs`. One may wonder why we consider a perturbation mechanism with three outputs (i.e., `Three-Outputs`) instead of a perturbation mechanism with four outputs (referred to as `Four-Outputs`), since using two bits to encode the output of a perturbation mechanism can represent four outputs. The reason is as follows. The approach to design `Four-Outputs` is similar to that for `Three-Outputs`, but the detailed analysis for `Four-Outputs` will be even more tedious than that for `Three-Outputs` (which is already quite complex). Given above reasons, we elaborate `Three-Outputs` but not `Four-Outputs` in this chapter.

3.3.2 `PM-OPT` Mechanism

Now, we advocate an optimal piecewise mechanism (`PM-OPT`) as shown in Algorithm 3 to get a small worst-case variance when the privacy budget is large. As shown in Fig. 3.2, `Three-Outputs`' worst-case noise variance is smaller than `PM`'s

when the privacy budget $\epsilon < 3.2$. But it loses the advantage when the privacy budget $\epsilon \geq 3.2$. As the privacy budget increases, Kairouz *et al.* [144] suggested sending more information using more output possibilities. Besides, we observe that it is possible to improve Wang *et al.*'s [18] `PM` to achieve a smaller worst-case noise variance. Thus, inspired by them, we propose an optimal piecewise mechanism named as `PM-OPT` with a smaller worst-case noise variance than `PM`.

Algorithm 3: `PM-OPT` Mechanism for One-Dimensional Numeric Data under Local Differential Privacy.

Input: tuple $x \in [-1, 1]$ and privacy parameter ϵ.
Output: tuple $Y \in [-A, A]$.
Value t is calculated in the Eq. (3.29);
Sample u uniformly at random from $[0, 1]$;
if $u < \frac{e^\epsilon}{t+e^\epsilon}$ **then**
  Randomly generate a sample Y uniformly from $[L(\epsilon, x, t), R(\epsilon, x, t)]$;
else
  Randomly generate a sample Y uniformly from $[-A, L(\epsilon, x, t)) \cup (R(\epsilon, x, t), A]$;
return Y;

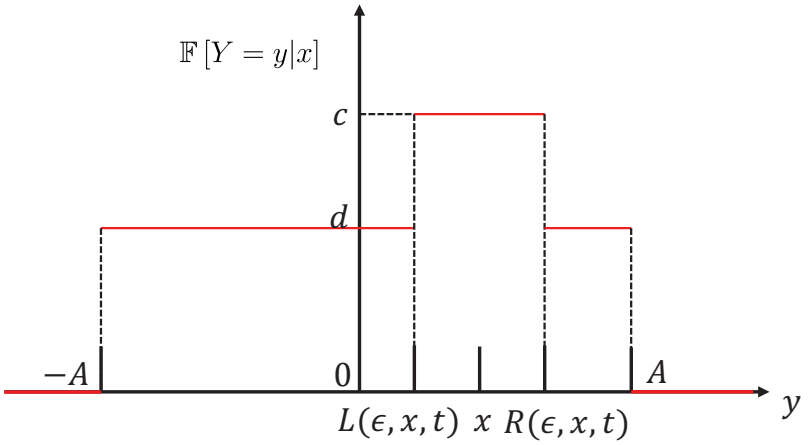


FIGURE 3.4: The probability density function $\mathbb{F}[Y = y|x]$ of the randomized output Y after applying ϵ-local differential privacy.

For a true input $x \in [-1, 1]$, the probability density function of the randomized output $Y \in [-A, A]$ after applying LDP is

$$\mathbb{F}[Y = y|x] = \begin{cases} c, & \text{for } y \in [L(\epsilon, x, t), R(\epsilon, x, t)], \quad (3.25a) \\ d, & \text{for } y \in [-A, L(\epsilon, x, t)) \cup (R(\epsilon, x, t), A], \quad (3.25b) \end{cases}$$

where

$$c = \frac{e^{\epsilon} t(e^{\epsilon}-1)}{2(t+e^{\epsilon})^2}, \tag{3.26}$$

$$d = \frac{t(e^{\epsilon}-1)}{2(t+e^{\epsilon})^2}, \tag{3.27}$$

$$A = \frac{(e^{\epsilon}+t)(t+1)}{t(e^{\epsilon}-1)}, \tag{3.28}$$

$$L(\epsilon, x, t) = \frac{(e^{\epsilon}+t)(xt-1)}{t(e^{\epsilon}-1)},$$

$$R(\epsilon, x, t) = \frac{(e^{\epsilon}+t)(xt+1)}{t(e^{\epsilon}-1)}, \text{ and}$$

$$t = \begin{cases} \frac{1}{2}\sqrt{e^{2\epsilon}+2^{2/3}\sqrt[3]{e^{2\epsilon}-e^{4\epsilon}}}+ \\ \frac{1}{2}\sqrt{2e^{2\epsilon}-2^{2/3}\sqrt[3]{e^{2\epsilon}-e^{4\epsilon}}+\frac{4e^{\epsilon}-2e^{3\epsilon}}{\sqrt{e^{2\epsilon}+2^{2/3}\sqrt[3]{e^{2\epsilon}-e^{4\epsilon}}}}} \\ -\frac{e^{\epsilon}}{2}, \quad \text{if } \epsilon < \ln\sqrt{2}, & (3.29a) \\ -\frac{1}{2}\sqrt{e^{2\epsilon}+2^{2/3}\sqrt[3]{e^{2\epsilon}-e^{4\epsilon}}}+ \\ \frac{1}{2}\sqrt{2e^{2\epsilon}-2^{2/3}\sqrt[3]{e^{2\epsilon}-e^{4\epsilon}}-\frac{4e^{\epsilon}-2e^{3\epsilon}}{\sqrt{e^{2\epsilon}+2^{2/3}\sqrt[3]{e^{2\epsilon}-e^{4\epsilon}}}}} \\ -\frac{e^{\epsilon}}{2}, \quad \text{if } \epsilon > \ln\sqrt{2}, & (3.29b) \\ \frac{\sqrt{3+2\sqrt{3}}-1}{\sqrt{2}}, \quad \text{if } \epsilon = \ln\sqrt{2}. & (3.29c) \end{cases}$$

The meaning of t can be seen from $\frac{t-1}{t+1} = \frac{L(\epsilon,1,t)}{R(\epsilon,1,t)}$. When the input is $x = 1$, the length of the higher probability density function $\mathbb{F}[Y = y|x] = \frac{e^{\epsilon}t(e^{\epsilon}-1)}{2(t+e^{\epsilon})^2}$ is $R(\epsilon,1,t) - L(\epsilon,1,t)$. $R(\epsilon,1,t)$ is the right boundary, and $L(\epsilon,1,t)$ is the left boundary. If $0 < t < \infty$, we can derive $\lim_{t\to 0}\frac{t-1}{t+1} = -1$, meaning the right boundary is opposite to the left boundary if t is close to 0. Since $\lim_{t\to\infty}\frac{t-1}{t+1} = 1$, it means that the right boundary is equal to the left boundary when t is close to ∞.

Moreover, Fig. 3.4 illustrates that the probability density function of Eq. (3.25) contains three pieces. If $y \in [L(\epsilon,x,t), R(\epsilon,x,t)]$, the probability density function is equal to c which is higher than other two pieces $y \in [-A, L(\epsilon,x,t))$ and $y \in$

$(R(\epsilon, x, t), A]$. We calculate the probability of the variable Y falling in the interval $[L(\epsilon, x, t), R(\epsilon, x, t)]$ as $\mathbb{P}\left[L(\epsilon, x, t) \leq Y \leq R(\epsilon, x, t)\right] = \int_{L(\epsilon,x,t)}^{R(\epsilon,x,t)} c \; dY = \frac{e^{\epsilon}}{t+e^{\epsilon}}$.

Furthermore, we use the following lemmas to establish how we get the value t in Eq. (3.25).

Lemma 6. Algorithm 3 achieves ϵ-local differential privacy. When the input value is x, the algorithm outputs a noisy value Y with $\mathbb{E}[Y|x] = x$ and

$$\mathrm{Var}[Y|x] = \frac{t+1}{e^{\epsilon}-1}x^2 + \frac{(t+e^{\epsilon})\big((t+1)^3+e^{\epsilon}-1\big)}{3t^2(e^{\epsilon}-1)^2}. \tag{3.30}$$

Proof. The proof details are given in Appendix A.6. □

Thus, when $x = 1$, we obtain the worst-case noise variance as follows:

$$\max_{x\in[-1,1]} \mathrm{Var}[Y|x] = \frac{t+1}{e^{\epsilon}-1} + \frac{(t+e^{\epsilon})\big((t+1)^3+e^{\epsilon}-1\big)}{3t^2(e^{\epsilon}-1)^2}. \tag{3.31}$$

Then, we obtain the optimal t in Lemma 7 to minimize Eq. (3.31).

Lemma 7. The optimal t for $\min_t \max_{x\in[-1,1]} \mathrm{Var}[Y|x]$ is Eq. (3.29).

Proof. By computing the first-order derivative and second-order derivative of $\min_t \max_{x\in[-1,1]} \mathrm{Var}[Y|x]$, we get the optimal t. The proof details are given in Appendix A.7. □

3.3.3 PM-SUB Mechanism

We propose a suboptimal piecewise mechanism (PM-SUB) to simplify the sophisticated computation of t in Eq. (3.4) of PM-OPT, and details of PM-SUB are shown in Algorithm 4.

Fig. 3.2 illustrates that PM-OPT achieves a smaller worst-case noise variance compared with PM, but the parameter t for PM-OPT in Eq. (3.29) is complicated to compute. Some vehicles are unable to process the complicated computation. To make t simple for vehicles to implement, we need to find a simple expression for it while ensuring the mechanism's performance. Then, we find that Wang *et al.*'s [18] PM is the case when $t = e^{\epsilon/2}$. Inspired by PM, $\ln t$ and ϵ can be linearly related.

Algorithm 4: `PM-SUB` Mechanism for One-Dimensional Numeric Data under Local Differential Privacy.

Input: tuple $x \in [-1, 1]$ and the privacy parameter ϵ.
Output: tuple $Y \in [-A, A]$.
Randomly generate a sample u uniformly from $[0, 1]$;
if $u < \frac{e^{\epsilon}}{e^{\epsilon/3}+e^{\epsilon}}$ **then**
  Randomly generate a sample Y from $[\frac{(e^{\epsilon}+e^{\epsilon/3})(xe^{\epsilon/3}-1)}{e^{\epsilon/3}(e^{\epsilon}-1)}, \frac{(e^{\epsilon}+e^{\epsilon/3})(xe^{\epsilon/3}+1)}{e^{\epsilon/3}(e^{\epsilon}-1)}]$;
else
  Randomly generate a sample Y uniformly from $[-A, \frac{(e^{\epsilon}+e^{\epsilon/3})(xe^{\epsilon/3}-1)}{e^{\epsilon/3}(e^{\epsilon}-1)}) \cup (\frac{(e^{\epsilon}+e^{\epsilon/3})(xe^{\epsilon/3}+1)}{e^{\epsilon/3}(e^{\epsilon}-1)}, A]$;
return Y;

Then, we find that $\frac{\ln t}{\epsilon}$ is close to $\frac{1}{3}$ (t for `PM-OPT` in Eq. (3.29)), so we can set $e^{\epsilon/3}$ as t in Eq. (3.25) for a new mechanism named as `PM-SUB`. The probability of the variable Y falling in the interval $[L(\epsilon, x, e^{\epsilon/3}), R(\epsilon, x, e^{\epsilon/3})]$ is $\frac{e^{\epsilon}}{e^{\epsilon/3}+e^{\epsilon}}$, and we give the detail of proof in Appendix A.8.

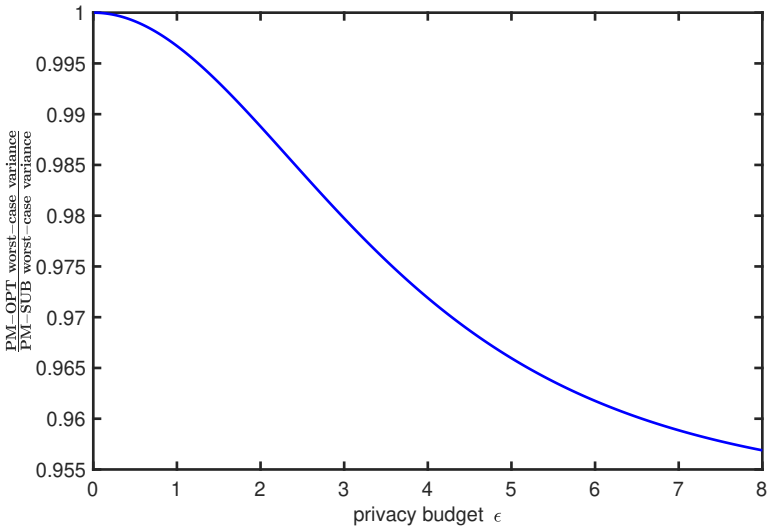


FIGURE 3.5: `PM-OPT`'s worst-case noise variance versus `PM-SUB`'s worst-case noise variance.

Similar to `PM-OPT`, we derive the worst-case noise variance of `PM-SUB` from Lemma 6 with $t = e^{\epsilon/3}$ as follows:

$$\max_{x\in[-1,1]} \mathrm{Var}[Y|x] = \frac{5e^{4\epsilon/3}}{3(e^{\epsilon}-1)^2} + \frac{5e^{2\epsilon/3}}{3(e^{\epsilon}-1)^2} + \frac{2e^{\epsilon}}{(e^{\epsilon}-1)^2}. \tag{3.32}$$

As shown in Fig. 3.5, `PM-SUB`'s worst-case noise variance is close to `PM-OPT`'s, but it is smaller than `PM`'s, which can be observed in Fig. 3.2.

3.3.4 Discretization Post-Processing

Both PM-OPT and PM-SUB's output ranges are $[-A, A]$ which is continuous, so that there are infinite output possibilities given an input x. Thus, it is difficult to encode their outputs for vehicles. Hence, we consider applying a post-processing process to discretize the continuous output range into finite output possibilities. Algorithm 5 shows our discretization post-processing steps.

Algorithm 5: Discretization Post-Processing.

Input: Perturbed data $y \in [-A, A]$, and domain $[-A, A]$ is separated into $2m$ pieces, where m is a positive integer.

Output: Discrete data Z.

Randomly generate a sample u from the Bernoulli distribution so that

$$\mathbb{P}[u = 1] = \left(\frac{A \cdot (\lfloor \frac{m \cdot y}{A} \rfloor + 1)}{m} - y \right) \cdot \frac{m}{A};$$

if $u = 1$ **then**

$\quad Z = \frac{A \cdot \lfloor \frac{m \cdot y}{A} \rfloor}{m}$;

else

$\quad Z = \frac{A \cdot (\lfloor \frac{m \cdot y}{A} \rfloor + 1)}{m}$;

return Z;

The idea of Algorithm 5 is as follows. We discretize the output range into $2m$ parts due to the symmetric range $[-A, A]$, and then we obtain $2m+1$ output possibilities. After we get a perturbed data y, it will fall into one of $2m$ segments. Then, we categorize it to the segment's left or right boundary, which resembles sampling a Bernoulli variable.

Next, we explain how we derive probabilities for the Bernoulli variable. Let the original input be x. A random variable Y represents the intermediate output after the perturbation and a random variable Z represents the output after the discretization. The range of Y is $[-A, A]$. Because the range of output is symmetric with respect to 0, we discretize both $[-A, 0]$ and $[0, A]$ into m parts, where the value of m depends on the user' requirement. Thus, we discretize Y to Z to take only the following $(2m + 1)$ values:

$$\left\{ i \times \frac{A}{m} : \text{integer } i \in \{-m, -m+1, \ldots, m\} \right\}. \tag{3.33}$$

When Y is instantiated as $y \in [-A, A]$, we have the following two cases:

① If y is one of the above $(2m+1)$ values, we set Z as y.

② If y is not one of the above $(2m+1)$ values, and then there exist some integer $k \in \{-m, -m+1, \ldots, m-1\}$ such that $\frac{kA}{m} < y < \frac{(k+1)A}{m}$. In fact, this gives $k < \frac{ym}{A} < k+1$, so we can set $k := \lfloor \frac{ym}{A} \rfloor$. Then conditioning on that Y is instantiated as y, we set Z as $\frac{kA}{m}$ with probability $k+1-\frac{ym}{A}$ and as $\frac{(k+1)A}{m}$ with probability $\frac{ym}{A}-k$, so that the expectation of Z given $Y=y$ equals y (as we will show in Eq. (A.107), this ensures that the expectation of Z given the original input as x equals x).

The following Lemma 8 shows the probability distribution of assigning y with a boundary value in the second case above when the intermediate output y is not one of discrete $(2m+1)$ values.

Lemma 8. After we obtain the intermediate output y after perturbation, we discretize it to a random variable Z equal to $\frac{kA}{m}$ or $\frac{(k+1)A}{m}$ with the following probabilities:

$$\mathbb{P}\left[Z = z \mid Y = y\right] = \begin{cases} k+1-\frac{ym}{A}, & \text{if } z = \frac{kA}{m}, \\ \frac{ym}{A}-k, & \text{if } z = \frac{(k+1)A}{m}. \end{cases} \tag{3.34}$$

Proof. The proof details are given in Appendix A.9. □

After discretization, the worst-case noise variance does not change or get worse proved by Lemma 9 as follows:

Lemma 9. Let local differential privacy mechanism be Mechanism $\mathcal{M}_1$, and discretization algorithm be Mechanism $\mathcal{M}_2$. Let all of the output possibilities of Mechanism $\mathcal{M}_1$ be S_1, and output possibilities of Mechanism $\mathcal{M}_2$ be S_2. $S_2 \subset S_1$. When given input x, $\mathcal{M}_1$ and $\mathcal{M}_2$ are unbiased. The worst-case noise variance of Mechanism $\mathcal{M}_2$ is greater than or equal to that of Mechanism $\mathcal{M}_1$.

Proof. The proof details are given in Appendix A.10. □

3.3.5 HM-TP Mechanism

Fig. 3.2 shows that Three-Outputs outperforms PM-SUB when the privacy budget ϵ is small, whereas PM-SUB achieves a smaller variance if the privacy budget ϵ is large. To fully take advantage of two mechanisms, we combine Three-Outputs and PM-SUB to create a new hybrid mechanism named as HM-TP. Fig. 3.2 illustrates that HM-TP obtains a lower worst-case noise variance than other solutions.

Hence, HM-TP invokes PM-SUB with probability β. Otherwise, it invokes Three-Outputs. We define the noisy variance of HM-TP as $\mathrm{Var}_{\mathcal{H}}[Y|x]$ given inputs x as follows:

$$\mathrm{Var}_{\mathcal{H}}[Y|x] = \beta \cdot \mathrm{Var}_{\mathcal{P}}[Y|x] + (1-\beta) \cdot \mathrm{Var}_{\mathcal{T}}[Y|x],$$

where $\mathrm{Var}_{\mathcal{P}}[Y|x]$ and $\mathrm{Var}_{\mathcal{T}}[Y|x]$ denote noisy outputs' variances incurred by PM-SUB and Three-Outputs, respectively. The following lemma presents the value of β:

Lemma 10. The $\max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x]$ is minimized when β is Eq. (A.125). Due to the complicated equation of β, we put it in the appendix.

Proof. The proof details are given in Appendix A.14. □

Since we have obtained the probability β, we can calculate the exact expression for the worst-case noise variance in Lemma 11 as follows:

Lemma 11. If β satisfies Lemma 10, we obtain the worst-case noise variance of HM-TP as

$$\max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x] = \begin{cases} \mathrm{Var}_{\mathcal{H}}[Y|x^*], \text{if } 0 < \beta < \frac{2(e^\epsilon-a)^2(e^\epsilon-1)-ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon-a)^2(e^\epsilon+t)-ae^\epsilon(e^\epsilon+1)^2}, \\ \max\{\mathrm{Var}_{\mathcal{H}}[Y|0], \mathrm{Var}_{\mathcal{H}}[Y|1]\}, \quad \text{otherwise}, \end{cases}$$

where $x^* := \frac{(\beta-1)ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon-a)^2(\beta(e^\epsilon+t)-e^\epsilon+1)}$ and $a = P_{0\leftarrow 0}$ which is defined in Eq. (3.4).

Proof. The proof details are given in Appendix A.11. □

3.4 Mechanisms for Estimation of Multiple Numeric Attributes

We consider a case in which the user's data record contains $d > 1$ attributes. There are three existing solutions to collect multiple attributes: (i) The straightforward approach collects each attribute with privacy budget ϵ/d. Based on the composition theorem [8], it satisfies ϵ-LDP after collecting all attributes. But the added noise is excessive if d is large [18]. (ii) `Duchi` *et al.*'s solution [63], which is somewhat complicated, handles numeric attributes only. (iii) Wang *et al.*'s solution [18] is the advanced approach that deals with a data tuple containing both numeric and categorical attributes. Their algorithm requires calculating an optimal $k < d$ based on the single-dimensional attribute's ϵ-LDP mechanism, and a user submits selected k dimensional attributes instead of d dimensions.

Algorithm 6: Mechanism for Multiple-Dimensional Numeric Attributes.

Input: tuple $x \in [-1, 1]^d$ and privacy parameter ϵ.

Output: tuple $Y \in [-A, A]^d$.

Let $Y = \langle 0, 0, \ldots, 0 \rangle$;

Let $k = \max\{1, \min\{d, \left\lfloor \frac{\epsilon}{2.5} \right\rfloor\}\}$;

Randomly generate k samples uniformly without replacement from $\{1, 2, \ldots, d\}$;

for *each sampled value* j **do**

  Input $x[t_j]$ and $\frac{\epsilon}{k}$ to `PM-SUB`, `Three-Outputs` or `HM-TP`, and output a noisy value y_j for each mechanism;

  $Y[t_j] = \frac{d}{k} y_j$;

return Y;

Thus, we follow Wang *et al.*'s [18] idea to extend Section 3.3 to the case of multidimensional attributes. The pseudo-code of our extension for our `PM-SUB`, `Three-Outputs`, and `HM-TP` is shown in Algorithm 6. When fed in a tuple $x \in [-1, 1]^d$, the algorithm outputs a perturbed tuple Y which includes non-zero value on k attributes, where

$$k = \max\{1, \min\{d, \left\lfloor \frac{\epsilon}{2.5} \right\rfloor\}\}, \tag{3.35}$$

and Appendix A.20 proves our selected k is optimal after extending `PM-SUB`, `Three-Outputs`, and `HM-TP` to support d dimensional attributes.

Overall, our algorithm for collecting multiple attributes outperforms existing solutions, which is confirmed by our experiments in Section 3.5. But `Three-Outputs` uses only one more bit compared with `Duchi` *et al.*'s [63] solution to encode outputs. Moreover, our `Three-Outputs` obtains a higher accuracy when the privacy regime is high (the privacy budget is small) and saves many bits for encoding since `PM` and `HM`'s continuous output range requires infinite bits to encode, whereas `PM-SUB` and `HM-TP`'s advantages are obvious at a large privacy budget. Furthermore, we discretize the continuous range of outputs to discrete outputs because vehicles cannot encode continuous range. Our experiments in Section 3.5.3 confirm that we can achieve similar results to algorithms before discretizing by carefully designing the number of discrete parts. Hence, our proposed algorithms are more suitable for vehicles than existing solutions.

Intuitively, Algorithm 6 requires every user to submit k attributes instead of d attributes, such that each attribute's allocated privacy budget increases from ϵ/d to ϵ/k, which helps to minimize the noisy variance. In addition, by setting k as Eq. (3.35), Algorithm 6 achieves an asymptotically optimal performance while preserving privacy, which we will prove using Lemma 12 and 13. Lemma 12 and 13 are proved in the same way as that of Lemma 4 and 5 in [18].

Lemma 12. Algorithm 6 satisfies ϵ-local differential privacy. Additionally, when the input tuple is x, the output is a noisy tuple Y, such that for any $j \in [1, d]$, and each t_j of those k attributes which is selected uniformly (without replacement) from all d attributes of x, $\mathbb{E}[Y[t_j]] = x[t_j]$ holds.

Proof. Algorithm 6 composes k numbers of ϵ-LDP perturbation algorithms; thus, based on composition theorem of differential mechanism [12], Algorithm 6 satisfies ϵ-LDP. As we can see from Algorithm 6, each perturbed output Y equals to $\frac{d}{k}y_j$ with probability $\frac{k}{d}$ or equals to 0 with probability $1 - \frac{k}{d}$. Thus, $\mathbb{E}[Y[t_j]] = \frac{k}{d} \cdot \mathbb{E}[\frac{d}{k} \cdot y_j] = \mathbb{E}[y_j] = x[t_j]$ holds. □

Lemma 13. For any $j \in [1, d]$, let $Z[t_j] = \frac{1}{n}\sum_{i=1}^{n} Y[t_j]$ and $X[t_j] = \frac{1}{n}\sum_{i=1}^{n} x[t_j]$. With at least $1 - \beta$ probability,

$$\max_{j \in [1,d]} |Z[t_j] - X[t_j]| = O\left(\frac{\sqrt{d \ln(d/\beta))}}{\epsilon\sqrt{n}}\right).$$

Proof. The proof details are given in Appendix A.19. □

3.5 Experiments

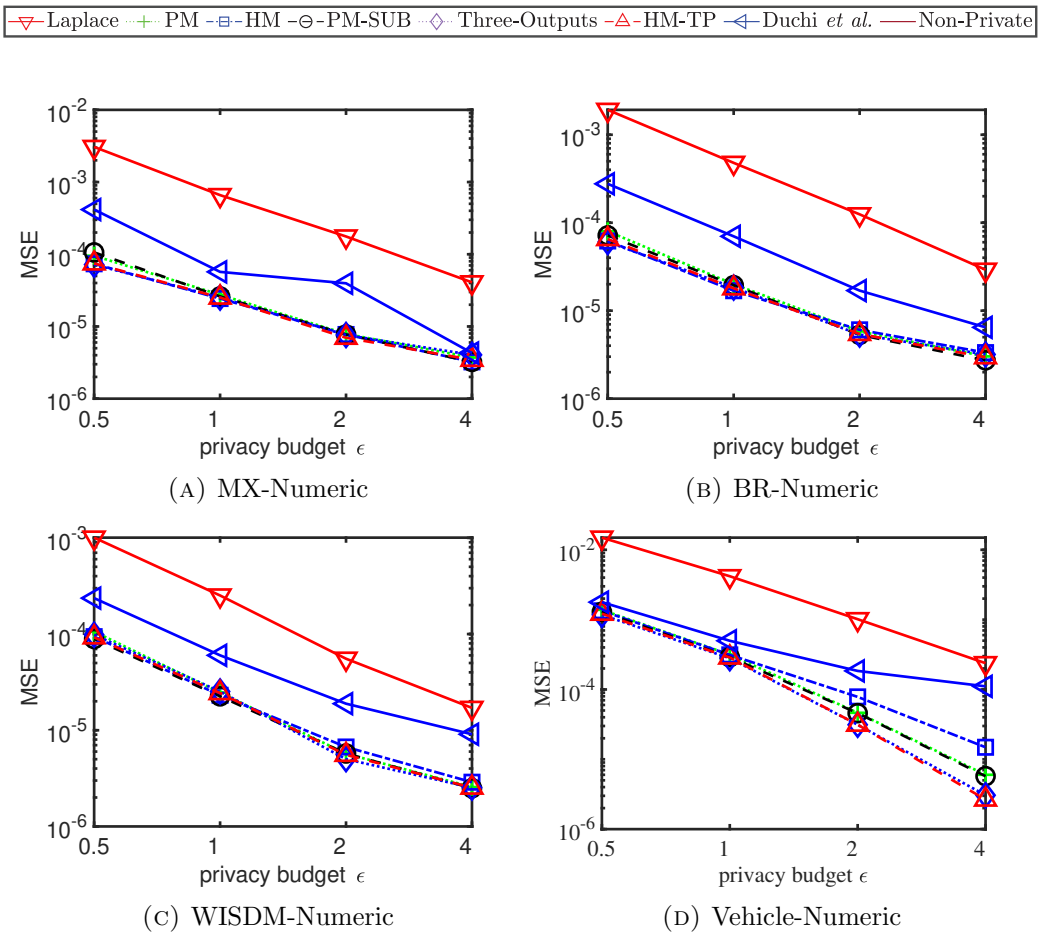
We implemented both existing solutions and our proposed solutions, including the `Laplace` mechanism, `Duchi` *et al.*'s [63] solution, `PM` and `HM` proposed by Wang *et al.* [18], `PM-SUB`, `Three-Outputs`, and `HM-TP`. Our datasets include (i) the WISDM Human Activity Recognition dataset [145] is a set of accelerometer data collecting on Android phones from 35 subjects performing 6 activities, where the domain of the timestamps of the phone's uptime is removed from the dataset, and the remaining 3 numeric attributes are accelerations in x, y, and z directions measured by the Android phone's accelerometer and 2 categorical attributes; (ii) two public datasets extracted from Integrated Public Use Microdata Series [146] contain census records from Mexico (MX) and Brazil (BR). MX contains $4M$ records and 19 attributes where 14 are categorical, and 5 are numerical. BR includes $4M$ tuples and 16 attributes, of which 10 are categorical, and 6 are numerical; (iii) a Vehicle dataset obtained by collecting from a distributed sensor network, including infrared (polarized IR sensor), acoustic (microphone), and seismic (geophone) [147]. The dataset contains $98,528$ tuples and 101 attributes, where 100 attributes represent information such as the raw time series data observed at each sensor and acoustic feature vectors extracted from each sensor's microphone. One attribute is categorical, denoting different types of vehicles, which are labeled manually by a human operator to ensure high accuracy. Besides, information about vehicles is gathered to find out the type or brand of the vehicle. The Vehicle dataset is also used as FL benchmark by [148]. We normalize every numeric attribute's domain to $[-1, 1]$.

3.5.1 Results on the Mean Values of Numeric Attributes

By collecting a noisy multidimensional tuple from each user, we estimate the mean of every numeric attribute. In order to compare with Wang *et al.*'s [18] mechanisms, we follow their experiments and then divide the total privacy budget ϵ into two parts. Assume a tuple contains d attributes which include d_n numeric attributes and d_c categorical attributes. Then, $d_n\epsilon/d$ budget is allocated to numeric attributes, and $d_c\epsilon/d$ to categorical ones. Our approach of using LDP for categorical data is the same as that of Wang *et al.* [18]. We estimate the mean value for

each of the numeric attributes using existing methods: (i) `Duchi` *et al.*'s [63] solution deals with multiple numeric attributes directly; (ii) when using the `Laplace` mechanism, it applies ϵ/d budget to each numeric attribute individually; (iii) `PM` and `HM` are from Wang *et al.* [18]. In Section 3.4, for numeric attributes, the mean square error (MSE) of the estimated mean values is evaluated using our proposed approaches. Fig. 3.6 presents MSE results as a function of the total budget of ϵ in the datasets (WISDM, MX, BR, and Vehicle). To simplify the complexity, we use the Vehicle dataset's last 6 numerical attributes to calculate MSE. Overall, our experimental evaluation shows that our proposed approaches outperform existing solutions. `HM-TP` outperforms existing solutions in all settings, whereas `PM-SUB`'s MSE is smaller than `PM`'s when privacy budget ϵ is large such as 4, and `Three-Outputs`' performance is better at a small privacy budget. Hence, experimental results are in accordance with our theories.

We also run a set of experiments on synthetic datasets that contain numeric attributes only. We create four synthetic datasets, including 16 numeric attributes whose values are sampled from the Gaussian distribution with standard deviation of $\frac{1}{4}$ and mean value $u \in \{0, \frac{1}{3}, \frac{2}{3}, 1\}$. Also, these values are truncated to $[-1, 1]$. By evaluating the MSE in estimating mean values of numeric attributes with our proposed mechanisms, we present our experimental results in Fig. 3.7. Hereby, we confirm that `PM-SUB`, `Three-Outputs`, and `HM-TP` outperform existing solutions.



(A) MX-Numeric (B) BR-Numeric

(C) WISDM-Numeric (D) Vehicle-Numeric

FIGURE 3.6: MSE for estimating mean values on numeric attributes.

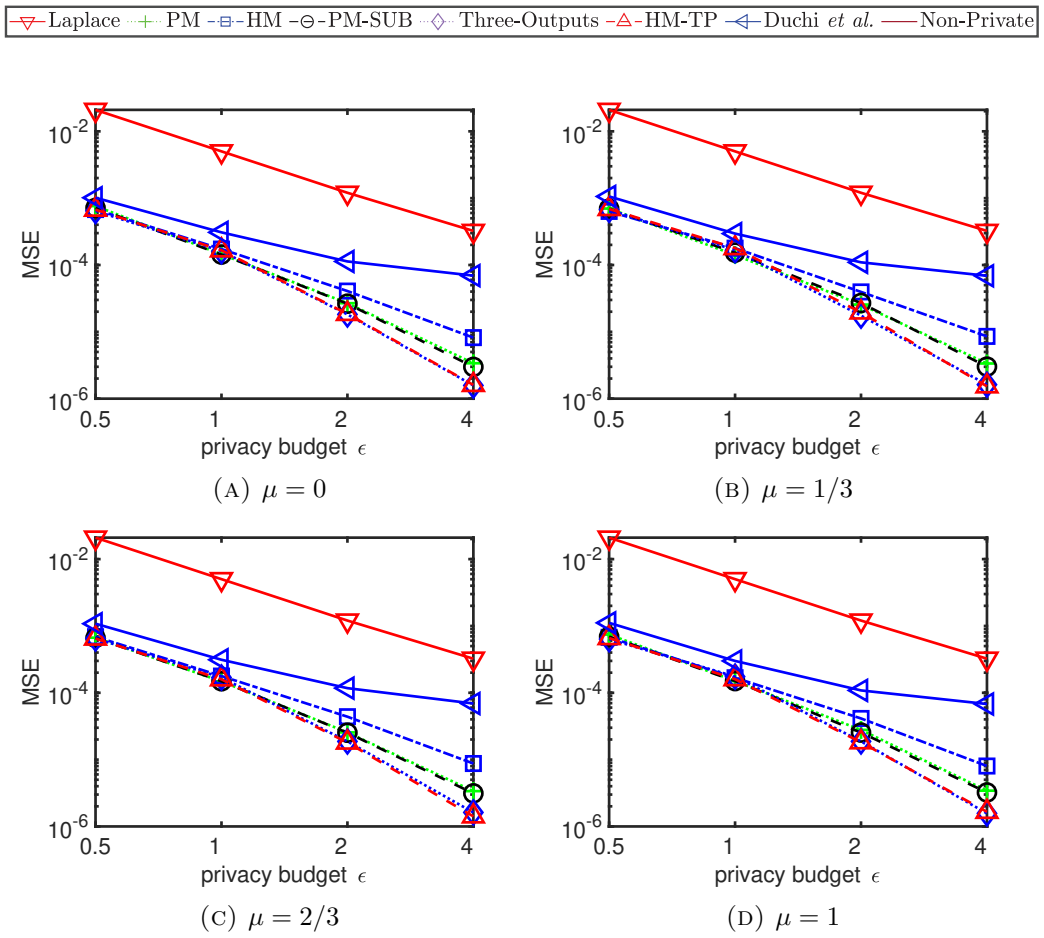


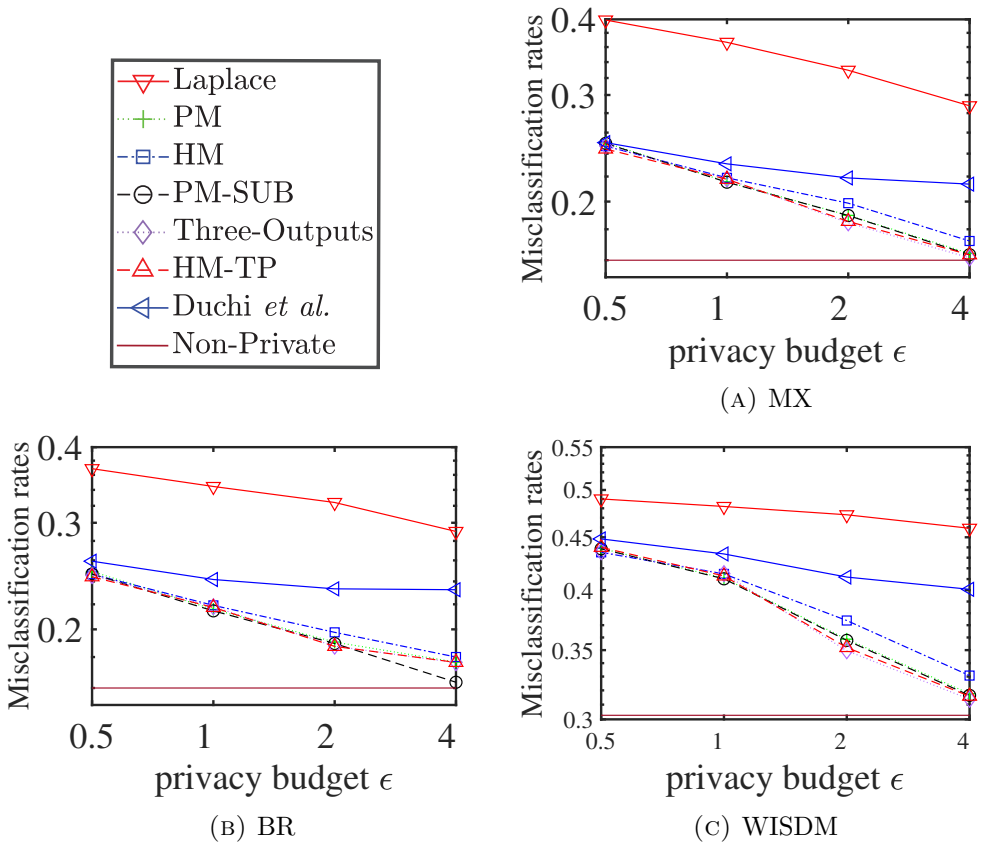
FIGURE 3.7: MSE for the estimated mean values (on synthetic datasets).

3.5.2 Results on Empirical Risk Minimization

The following experiments compare the proposed LDP mechanisms' performance by perturbing gradients generated in logistic regression, SVM classification, and linear regression tasks. We convert the categorical attribute t_j with k values into $k-1$ binary attributes with a domain $\{-1, 1\}$, for example, given t_j, (i) 1 represents the l-th ($l < k$) value on the l-th binary attribute and -1 on each of the rest of $k-2$ attributes; (ii) on all binary attributes, -1 represents the k-th value. The dimension of WISDM becomes 43, BR (resp. MX) is 90 (resp. 94), and Vehicle is 101 after the transformation. Since both the BR and MX datasets contain the "total income" attribute, we consider it as the dependent variable and other variables as independent variables. The Vehicle dataset is used for SVM [147, 148].

Each tuple in the Vehicle dataset contains a 100-dimensional feature and a binary label.

Consider each tuple of data as the dataset of a vehicle, so vehicles calculate gradients and run different LDP mechanisms to generate noisy gradients. Each mini-batch is a group of vehicles. Thus, the centralized aggregator, i.e., the cloud server, updates the model after each group of vehicles send noisy gradients. The experiment involves 8 competitors: `Laplace` mechanism, `Duchi` *et al.*'s solution, `PM`, `HM`, `PM-SUB`, `Three-Outputs`, `HM-TP`, and a non-private setting. The regularization factor is set as $\lambda = 10^{-4}$ in all approaches. Each method performs 10-fold cross-validation 5 times on each dataset in our experiments. Fig. 3.8 and Fig. 3.10 show that the proposed mechanisms (`PM-SUB`, `Three-Outputs`, and `HM-TP`) have lower misclassification rates than other mechanisms. Fig. 3.9 shows the MSE of the linear regression model. We ignore Laplace's result because its MSE exceeds those of other mechanisms. In the selected privacy budgets, our proposed mechanisms (`PM-SUB`, `Three-Outputs`, and `HM-TP`) outperform existing approaches, including `Laplace` mechanism, `Duchi` *et al.*'s solution, `PM`, and `HM`.

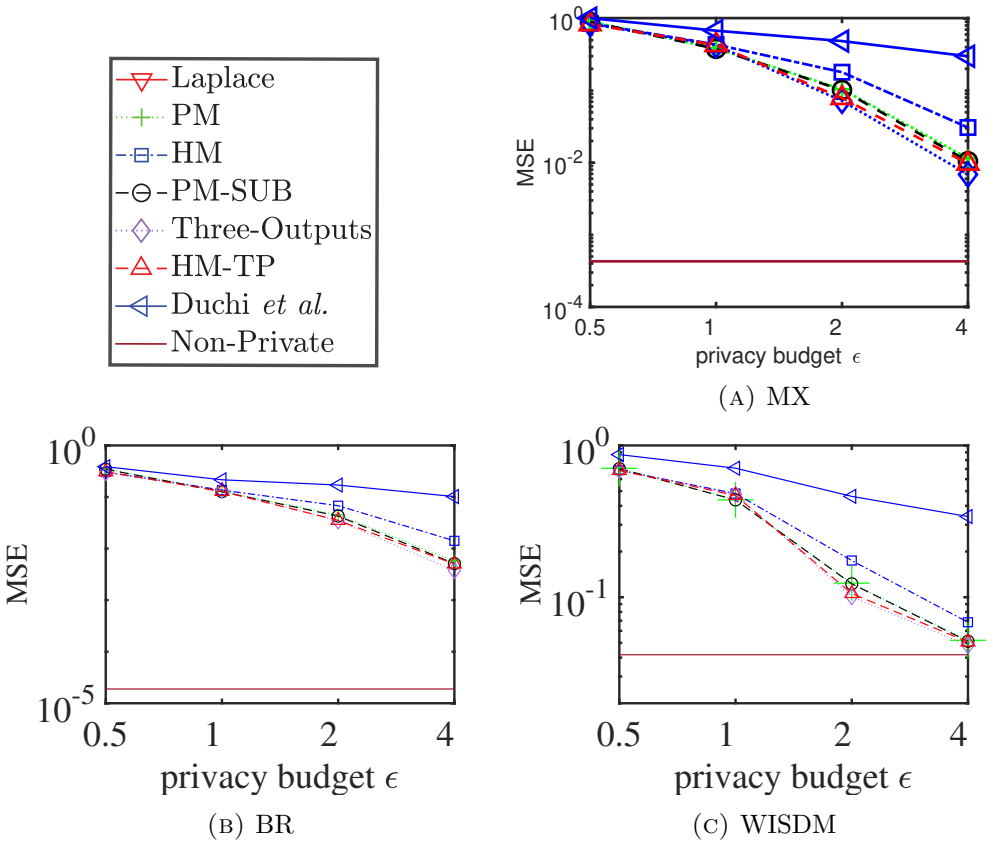


(A) MX

(B) BR

(C) WISDM

FIGURE 3.8: Logistic Regression.

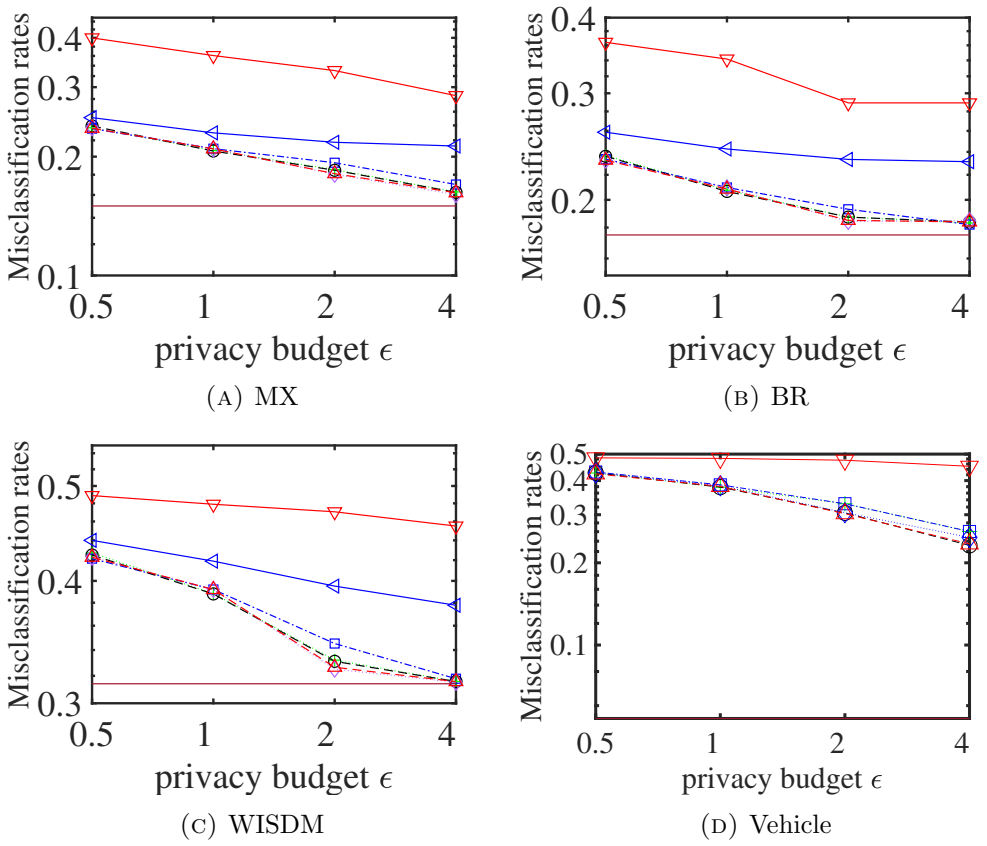


(A) MX

(B) BR

(C) WISDM

FIGURE 3.9: Linear Regression.



(A) MX (B) BR

(C) WISDM (D) Vehicle

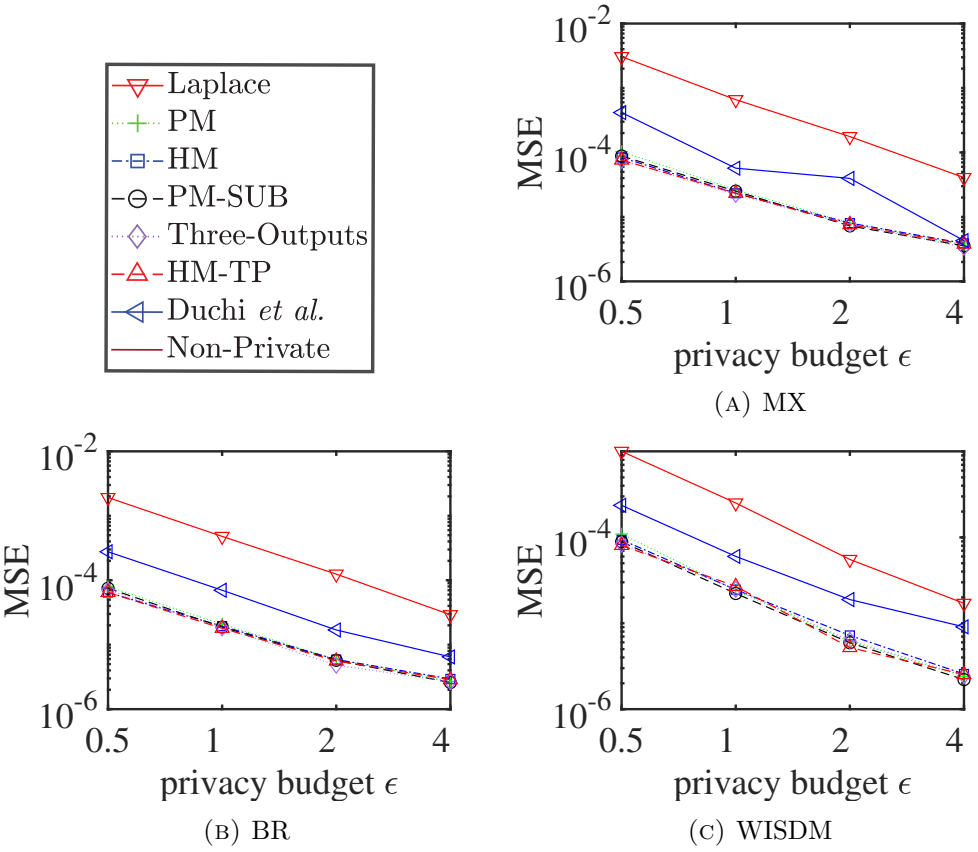
FIGURE 3.10: Support Vector Machines.

3.5.3 Results after Discretization

In this section, we add a discretization post-processing step in Algorithm 5 to the implementation of mechanisms with continuous range of outputs, including `PM`, `PM-SUB`, `HM` and `HM-TP`. To confirm that the discretization is effective, we perform the following experiments. We separate the output domain $[-A, A]$ into 2000 segments, and then we have 2001 possible outputs given an initial input x. We add a discretization step to the experiments in Section 3.5.1. Fig. 3.11 displays our experimental results. After discretizing, we confirm that our proposed approaches outperform existing solutions in estimating the mean value using three real-world datasets: WISDM, MX, and BR.

In addition, we use log regression and linear regression to evaluate the performance after discretization. We repeat the experiments in Section 3.5.2 with an additional discretization post-processing step. Fig. 3.12 and Fig. 3.13 show our experimental results. Compared with other approaches, the performance is similar to that

before discretizing. Furthermore, Fig. 3.14 illustrates how the accuracy changes as output possibilities increase. It shows that the misclassification rate of the logistic regression task and the MSE of the linear regression task are related to the size of output possibilities. Although incurring with randomness, we find that the misclassification rate and MSE decrease as the number of output possibilities increases. When there are three output possibilities, it incurs randomness. Moreover, Fig. 3.15 shows that `PM-SUB` outperforms `Three-Outputs`, when the number of output possibilities is large. However, when we discretize the range of outputs into 2000 segments, the performance is satisfactory and similar to a continuous range of outputs. Hence, our proposed approaches combined with the discretization step help retain the performance while enabling vehicle usage.

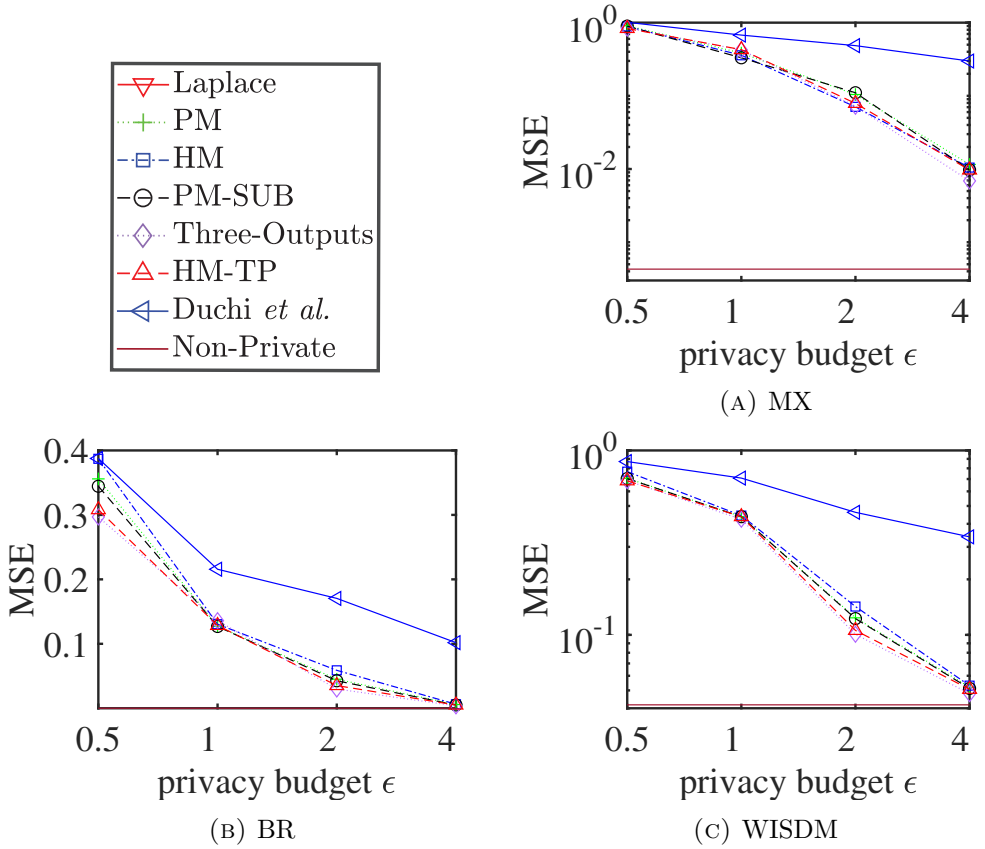


(A) MX

(B) BR

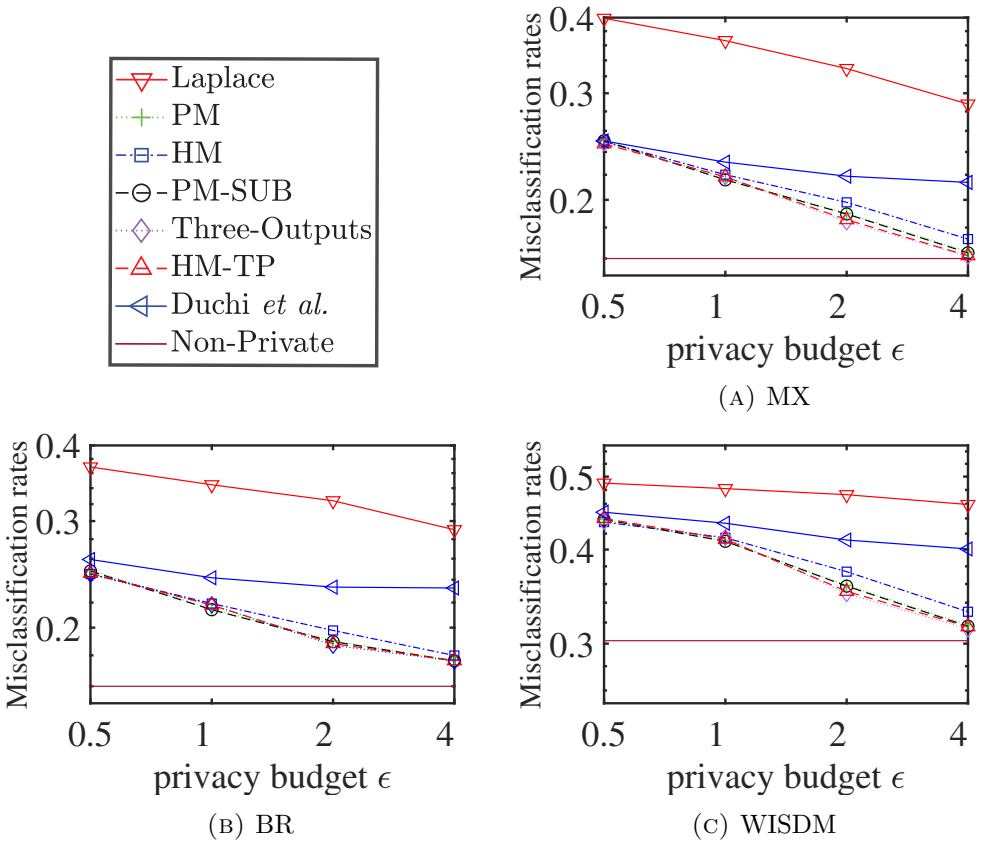
(C) WISDM

FIGURE 3.11: Result accuracy for mean estimation with discretization post processing on `PM`, `HM`, and `HM-TP`.



(A) MX (B) BR (C) WISDM

FIGURE 3.12: Linear Regression with discretization post processing on `PM`, `HM`, and `HM-TP` (privacy parameter $\epsilon = 4$).

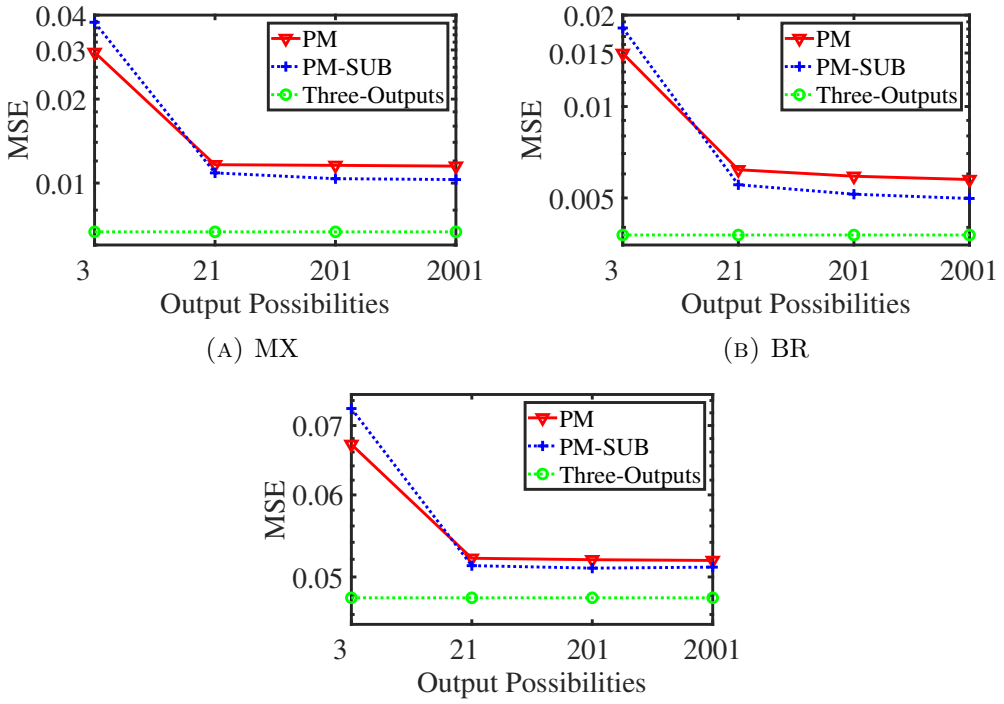


(A) MX

(B) BR

(C) WISDM

FIGURE 3.13: Logistic Regression with discretization post processing on `PM`, `HM`, and `HM-TP` (privacy budget $\epsilon = 4$).



(A) MX (B) BR

(C) WISDM

FIGURE 3.14: Linear Regression with discretization post processing on `PM`, `HM`, and `HM-TP` (privacy budget $\epsilon = 4$).

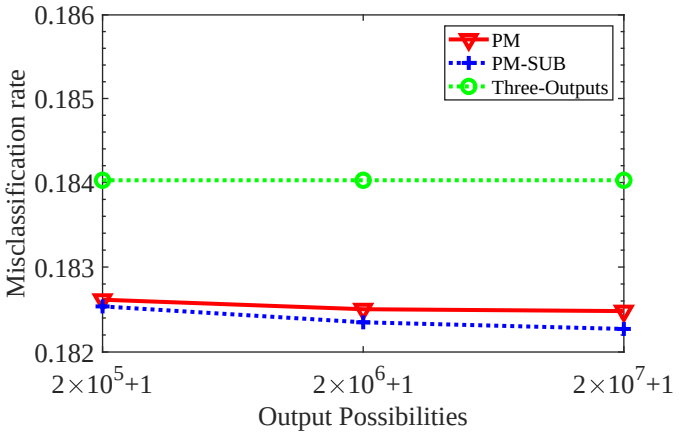


FIGURE 3.15: Support Vector Machine with discretization post processing on `PM`, `HM`, and `HM-TP` (privacy budget $\epsilon = 5$).

3.6 Summary

In this chapter, we propose `PM-OPT`, `PM-SUB`, `Three-Outputs`, and `HM-TP` LDP mechanisms. These mechanisms effectively preserve privacy when collecting data records and computing accurate statistics in various data analysis tasks, including estimating the mean frequency and machine learning tasks. Moreover, we integrate our proposed LDP mechanisms with `FedSGD` algorithm to create an `LDP-FedSGD` algorithm. The `LDP-FedSGD` algorithm enables the vehicular crowdsourcing applications to train a machine learning model to predict the traffic status while

avoiding the privacy threat and reducing the communication cost. More specifically, by leveraging LDP mechanisms, adversaries cannot deduce the exact location information of vehicles from uploaded gradients. Then, FL enables vehicles to train their local machine learning models with collected data and then send noisy gradients instead of data to the cloud server to obtain a global model. Extensive experiments demonstrate that our proposed approaches are effective and able to perform better than existing solutions.

Chapter 4

A Blockchain-Based Approach for Saving and Tracking Differential-Privacy Cost[1] [2]

The privacy budget and privacy loss measure the risk of privacy leakage for users who use the application. If the privacy loss exceeds the privacy budget, there is no privacy protection. Apple announces to apply differential privacy technology in their Emojis, New words, Deeplinks and Lookup Hints applications [14, 149]. They claim that the privacy loss is 1 or 2 for each datum submitted to its servers, but research shows that the privacy loss is as high as 16 per day [14]. Since the privacy budget is limited, we are inspired to develop an application to reduce the waste on the differential privacy budget.

A blockchain-based system to manage the differential privacy costs is investigated in this chapter. Also, the proposed algorithm is applicable to reuse the previous differential privacy costs and noises to save the differential privacy budget spending. The major contributions of this chapter are summarized as follows:

[1]The work in this chapter has been published as Yang Zhao, Jun Zhao, Jiawen Kang, Zehang Zhang, Dusit Niyato, Shuyu Shi, and Kwok-Yan Lam. "A Blockchain-Based Approach for Saving and Tracking Differential-Privacy Cost", in *IEEE Internet of Things Journal, DOI: 10.1109/JIOT.2021.3058209, 2020.*

[2]Leong Mei Han, Yang Zhao, and Jun Zhao. "POSTER: Blockchain-Based Differential Privacy Cost Management System." *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020.*

- First, a novel privacy-preserving algorithm with rigorous mathematical proof is designed to minimize accumulated privacy costs under a limited privacy budget by reusing previous noisy responses if the same query is received. Thus, a dataset can be used to answer more queries while preventing privacy leakage, which is essential for the datasets with frequent queries, e.g., medical record datasets.

- Second, our designed approach reduces the number of times to request the server significantly by taking advantage of recorded noisy results. Furthermore, we design and implement a blockchain-based system for tracking and saving DP costs. As a result, the dataset owner can know how the dataset has been used and be confident that no new privacy cost will be incurred for answering queries once the specified privacy budget is exhausted.

- Third, we implement the proposed system and algorithm according to a detailed sequence diagram and conduct experiments using a real-world dataset. Numerical results demonstrate that our proposed system and algorithm effectively save the privacy costs while keeping accuracy.

- Forth, by combining blockchain-based system with the algorithm, a data owner can host datasets locally while opening access to others in a privacy-preserving mode. Data owners can set a privacy budget and multiple query types, and then the blockchain smart contract will record every request, the associated privacy cost, and the noisy response. Unlike calling the data hosting server every time in naive solutions, our approach reduces the number of times to request the server significantly by taking advantage of recorded noisy results.

The rest of the chapter is organized as follows. Section 4.1 presents the system design including our proposed noise reuse algorithm. Section 4.2 describes challenges in implementing our system. In Section 4.3, we discuss experimental results to validate the effectiveness of our system. Section 4.4 concludes this chapter and identifies future directions.

Notation. Throughout this chapter, $\mathbb{P}[\cdot]$ denotes the probability, and $\mathbb{F}[\cdot]$ represents for the probability density function. The notation $\mathcal{N}(0, A)$ denotes a Gaussian random variable with zero mean and variance A, and means a fresh Gaussian noise

when it is used to generate a noisy query response. Notations used in the rest of this chapter are summarized in Table 4.1.

TABLE 4.1: Summary of notations

(ϵ, δ)	privacy parameters
$\mathbb{P}[\cdot]$	probability
$\mathbb{F}[\cdot]$	probability density function
D	dataset D
D'	neighbouring dataset of D
Δ_Q	ℓ_2-sensitivity of query Q
σ	standard deviation
$Q_m(D)$	true query response for query Q_m on dataset D
$\widetilde{Q}_m(D)$	noisy query response for query Q_m on dataset D
$\mathcal{M}_{G1}, \mathcal{M}_{G2}, \ldots, \mathcal{M}_{Gm}$	randomized mechanisms $\mathcal{M}_{G1}, \mathcal{M}_{G2}, \ldots, \mathcal{M}_{Gm}$
r	the fraction of noise to be reused in a previous noisy response to generate the new noisy response
$L_{\mathcal{M}_G}(D, D'; y)$	privacy loss for the mechanism $\mathcal{M}_G$ with respect to neighbouring datasets D, D' when the output is y
$\mathcal{N}(0, \mathcal{A})$	a Gaussian random variable with zero mean and variance $\mathcal{A}$
V	variance
t	query type t
$\mathbf{\Sigma}_t$	the noise amounts for previous instances of type t-query

4.1 System Description

Our blockchain-based system provides differentially private responses to queries while minimizing the privacy costs via noise reuse. We design a web application to implement our Algorithm 7, which generates noisy responses to queries with the minimal privacy costs by setting the optimal reuse fraction of the old noisy response and adding new noise (if necessary). For clarity, we defer Algorithm 7 and its discussion to Section 4.1.4. The design of the system is shown in Fig. 4.1. Then, we discuss the implementation and experiments of our blockchain-based system, and present more figures about the results in Section 4.2 and Section 4.3.

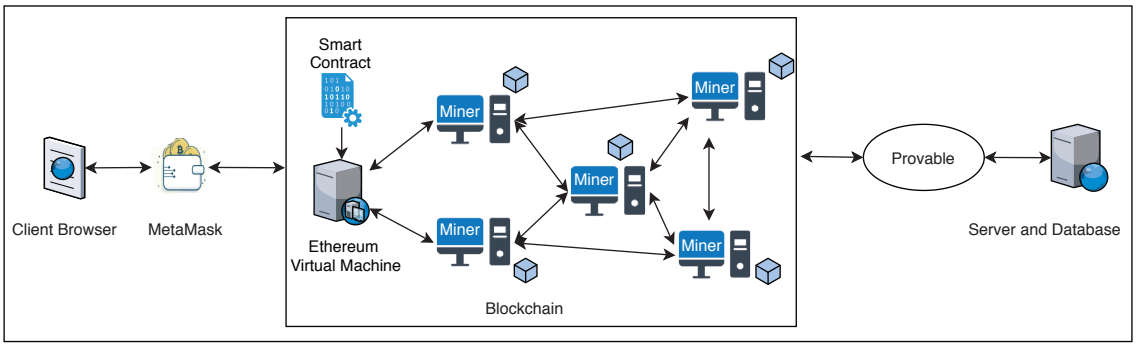


FIGURE 4.1: The proposed blockchain-based system architecture for differential-privacy costs management.

In particular, Fig. 4.3 there shows the screenshot of our blockchain-based privacy management system [150], while Fig. 4.4 presents outputs while using the system.

4.1.1 System Architecture

Our system includes the client, the blockchain, the server, and the smart contract followed by more details as below.

Client: The primary function of the client is to transfer users' queries to the blockchain smart contract. The client computes the required parameter standard deviation for the server to generate the Gaussian noise using the privacy parameters ϵ and δ and forwards the query to the blockchain. Also, the client can display the query result to the analyst after getting the noisy response to the query.

Blockchain Smart Contract: The blockchain serves as a middleware between the client and the server. It decides which query should be submitted to the server. The blockchain records the remaining privacy budget, query type, the noisy response to answer the query, the privacy parameters, and the amount of corresponding noise. If the remaining privacy budget is enough, the smart contract will execute the query match function with the recorded history. Otherwise, the smart contract will reject this query. If the current query does not match with any query in the history, the smart contract will call the server to calculate the result. If the query has been received before, the blockchain smart contract will not call the server if the noisy response can be completely generated by old noisy answers and will call the server if the access to the dataset is still needed to generate the noisy response.

Server: The data provider hosts the server. The server provides APIs to answer analysts' queries. When the API is called, the server will query the dataset to calculate the respective answer. After the true value $Q(D)$ is calculated, the server will add noise to perturb the answer. Then the server returns the noisy answer to the blockchain.

In the rest of the chapter, we utilize `Blockchain`, `Client`, and `Server` to denote the blockchain, client, and server, respectively.

4.1.2 System Functionality

Match query with query history and generate noisy response: `Blockchain` compares the current query type with saved query types to retrieve previous query results. If it is the first time for `Blockchain` to see the query, `Blockchain` will forward the query to the server, and `Server` will return the perturbed result which satisfies differential privacy to `Blockchain`. If the current query type matches previous answers' query type, `Blockchain` will compare the computed amount of noise with all previously saved amounts of noise under the same query type. Based on the comparison result, `Blockchain` will completely reuse old responses or call `Server`.

Manage privacy budget: `Blockchain` updates the privacy budget as queries are answered, and `Blockchain` ensures no new privacy costs will be incurred for answering queries once the specified privacy budget is exhausted.

4.1.3 Adversary Model

The adversary model for our system is similar to [43]. Assume that there are two kinds of adversaries:

First, adversaries can obtain perturbed query results. They may try to infer users' real information using perturbed queries' results.

Second, adversaries attempt to modify the privacy budget. For example, they would like to decrease the used privacy budget so that users may exceed the privacy budget. As a result, privacy will leak. However, in our case, the privacy budget is recorded on the blockchain. The adversaries cannot tamper it once the privacy budget is stored in the blockchain.

Algorithm 7: Our proposed algorithm to answer the m-th query and adjust remaining privacy costs.

Input: D: dataset; Q_m: the m-th query; (ϵ_m, δ_m): requested privacy parameters for query Q_m; $(\sqrt{\epsilon\_\text{squared\_remaining\_budget}}, \delta_{\text{budget}})$: remaining privacy budget (at the beginning, it is $(\sqrt{\epsilon\_\text{squared\_budget}}, \delta_{\text{budget}})$ for $\epsilon\_\text{squared\_budget} = {\epsilon_{\text{budget}}}^2$); Δ_{Q_m}: ℓ_2 sensitivity of query Q_m;

Output: $\widetilde{Q}_m(D)$: noisy query response for query Q_m on dataset D under (ϵ_m, δ_m)-differential privacy;

$\sigma_m \leftarrow \text{Gaussian}(\Delta_{Q_m}, \epsilon_m, \delta_m)$; //***Comment:*** *From Lemma 2.1, it holds that* $\text{Gaussian}(\Delta_{Q_m}, \epsilon_m, \delta_m) := \sqrt{2 \ln \frac{1.25}{\delta_m}} \times \frac{\Delta_{Q_m}}{\epsilon_m}$.

if the query Q_m is seen for the first time **then**

  `Client` computes $\epsilon\_\text{squared\_cost}$ such that $\text{Gaussian}(\Delta_{Q_m}, \sqrt{\epsilon\_\text{squared\_cost}}, \delta_{\text{budget}}) = \sigma_m$;

  //***Comment:*** *This means* $\sqrt{2 \ln \frac{1.25}{\delta_{\text{budget}}}} \times \frac{\Delta_{Q_m}}{\sqrt{\epsilon\_\text{squared\_cost}}} = \sigma_m$, *where* σ_m *as* $\text{Gaussian}(\Delta_{Q_m}, \epsilon_m, \delta_m)$ *is* $\sqrt{2 \ln \frac{1.25}{\delta_m}} \times \frac{\Delta_{Q_m}}{\epsilon_m}$.

  `Client` computes $\epsilon\_\text{squared\_remaining\_budget} \leftarrow \epsilon\_\text{squared\_remaining\_budget} - \epsilon\_\text{squared\_cost}$;

  if $\epsilon\_\text{squared\_remaining\_budget} \geq 0$ **then**

    return $\widetilde{Q}_m(D) \leftarrow Q_m(D) + \mathcal{N}(0,1) \times \sigma_m$; //***Comment:*** *We refer to this Case 1) in Section 4.1.4. If Q_m is multidimensional, independent Gaussian noise will be added to each dimension.*

    `Blockchain` records $\langle Q_m\text{'s query type}, \epsilon_m, \delta_m, \sigma_m, \widetilde{Q}_m(D) \rangle$; //***Comment:*** *This information will be kept together with a cryptographic hash of the dataset D, which `Blockchain` stores so it knows which records are for the same dataset D.*

  else

    return an error of insufficient privacy budget;

  end if

else

  Suppose Q_m is a type t-query. `Blockchain` compares σ_m with values in $\mathbf{\Sigma}_t := \{\sigma_j : \sigma_j$ has been recorded in `Blockchain` and Q_j is a type t-query$\}$ (i.e., $\mathbf{\Sigma}_t$ consists of the corresponding noise amounts for previous instances of type t-query), resulting in the following subcases.

  if there exists $\sigma_j \in \mathbf{\Sigma}_t$ such that $\sigma_m = \sigma_j$ **then**

    `Blockchain` returns $\widetilde{Q}_m(D) \leftarrow \widetilde{Q}_j(D)$; //***Comment:*** *We refer to this Case 2A) in Section 4.1.4.*

  else if $\sigma_m < \min(\mathbf{\Sigma}_t)$ **then**

    //***Comment:*** ***The case of partially reusing an old noise***:

    `Client` computes $\epsilon\_\text{squared\_cost}$ such that $[\text{Gaussian}(\Delta_{Q_m}, \sqrt{\epsilon\_\text{squared\_cost}}, \delta_{\text{budget}})]^{-2} = {\sigma_m}^{-2} - [\min(\mathbf{\Sigma}_t)]^{-2}$;

    `Client` computes $\epsilon\_\text{squared\_remaining\_budget} \leftarrow \epsilon\_\text{squared\_remaining\_budget} - \epsilon\_\text{squared\_cost}$;

    if $\epsilon\_\text{squared\_remaining\_budget} \geq 0$ **then**

      `Blockchain` computes $\text{NoiseReuseRatio} \leftarrow \frac{{\sigma_m}^2}{[\min(\mathbf{\Sigma}_t)]^2}$ and $\text{AdditionalNoise} \leftarrow \mathcal{N}(0,1) \times \sqrt{{\sigma_m}^2 - \frac{{\sigma_m}^4}{[\min(\mathbf{\Sigma}_t)]^2}}$

      `Blockchain` contacts `Server` to compute $\widetilde{Q}_m(D) \leftarrow Q_m(D) + \text{NoiseReuseRatio} \times [\widetilde{Q}_{t,\min}(D) - Q_m(D)] + \text{AdditionalNoise}$, where $\widetilde{Q}_{t,\min}(D)$ denotes the noisy response (kept in `Blockchain`) corresponding to $\min(\mathbf{\Sigma}_t)$; //***Comment:*** *We refer to this Case 2B) in Section 4.1.4.*

      `Blockchain` records $\langle Q_m\text{'s query type}, \epsilon_m, \delta_m, \sigma_m, \widetilde{Q}_m(D) \rangle$;

    else

      return an error of insufficient privacy budget;

    end if

  else

    //***Comment:*** ***The case of fully reusing an old noise***:

    With σ_ℓ denoting the maximal possible value in $\mathbf{\Sigma}_t$ that is also smaller than σ_m, `Blockchain` reuses $\widetilde{Q}_\ell(D)$, which denotes the noisy response (kept in `Blockchain`) corresponding to σ_ℓ;

    `Blockchain` computes $\widetilde{Q}_m(D) \leftarrow \widetilde{Q}_\ell(D) + \mathcal{N}(0,1) \times \sqrt{{\sigma_m}^2 - {\sigma_\ell}^2}$; //***Comment:*** *We refer to this Case 2C) in Section 4.1.4.*

    `Blockchain` records $\langle Q_m\text{'s query type}, \epsilon_m, \delta_m, \sigma_m, \widetilde{Q}_m(D) \rangle$;

  end if

end if

4.1.4 Our Algorithm 7 based on Reusing Noise

We present our solution for reusing noise in Algorithm 7 in Section 4.1.5. We consider real-valued queries so that the Gaussian mechanism can be used. Extensions to non-real-valued queries can be regarded as future work, where we can apply the exponential mechanism of [12].

To clarify notation use, we note that Q_i means the i-th query (ordered chronologically) and is answered by a randomized algorithm $\widetilde{Q}_i$. A type t-query means that the query's type is t. Queries asked at different time can have the same query type. This is the reason that we reuse noise in Algorithm 7.

Suppose a dataset D has been used to answer $m-1$ queries $Q_1, Q_2, \ldots, Q_{m-1}$, where the i-th query Q_i for $i = 1, 2, \ldots, m-1$ is answered under (ϵ_i, δ_i)-differential privacy (by reusing noise, or generating fresh noise, or combining both). For $i = 1, 2, \ldots, m$, we define $\sigma_i := \text{Gaussian}(\Delta_{Q_i}, \epsilon_i, \delta_i)$, where Δ_{Q_i} denotes the ℓ_2-sensitivity of Q_i, where we defer the discussion of Δ_{Q_i} to Section 4.1.8. As presented in Algorithm 7, we have several cases discussed below. For better understanding of these cases, we later discuss an example given in Table 4.2 in Section 4.2.

Case 1): If Q_m is seen for the first time, we obtain the noisy response $\widetilde{Q}_m(D)$ by adding a zero-mean Gaussian noise with standard deviation $\text{Gaussian}(\Delta_{Q_m}, \epsilon_m, \delta_m)$ independently to each dimension of the true result $Q_m(D)$ (if the privacy budget allows), as given by Line 7 of Algorithm 7, where $\text{Gaussian}(\Delta_{Q_m}, \epsilon_m, \delta_m) := \sqrt{2 \ln \frac{1.25}{\delta_m}} \times \frac{\Delta_{Q_m}}{\epsilon_m}$ from Lemma 2.1.

Case 2): If Q_m has been received before, suppose Q_m is a type t-query, and among the previous $m-1$ queries $Q_1, Q_2, \ldots, Q_{m-1}$, let $\mathbf{\Sigma}_t$ consist of the corresponding noise amounts for previous instances of type t-query; i.e., $\mathbf{\Sigma}_t := \{\sigma_j : \sigma_j \text{ has been recorded in } \texttt{Blockchain} \text{ and } Q_j \text{ is a type } t\text{-query}\}$.

`Blockchain` compares σ_m and the values in $\mathbf{\Sigma}_t$, resulting in the following subcases.

Case 2A): If there exists $\sigma_j \in \mathbf{\Sigma}_t$ such that $\sigma_m = \sigma_j$, then $\widetilde{Q}_m(D)$ is set as $\widetilde{Q}_j(D)$.

Case 2B): This case considers that σ_m is less than $\min(\mathbf{\Sigma}_t)$ which denotes the minimum in $\mathbf{\Sigma}_t$. Let $\widetilde{Q}_{t,\min}(D)$ denote the noisy response (kept in

`Blockchain`) corresponding to $\min(\mathbf{\Sigma}_t)$; specifically, if $\min(\mathbf{\Sigma}_t) = \sigma_j$ for some j, then $\widetilde{Q}_{t,\min}(D) = \widetilde{Q}_j(D)$. Under $\sigma_m < \min(\mathbf{\Sigma}_t)$, to minimize the privacy costs, we reuse $\frac{{\sigma_m}^2}{[\min(\mathbf{\Sigma}_t)]^2}$ fraction of noise in $\widetilde{Q}_{t,\min}(D)$ to generate $\widetilde{Q}_m(D)$ (if the privacy budget allows). This will be obtained by Theorem 1's Result (ii) to be presented in Section 4.1.5. Specifically, under $\min(\mathbf{\Sigma}_t) > \sigma_m$, as given by Line 22 of Algorithm 7, $\widetilde{Q}_m(D)$ is set by $\widetilde{Q}_m(D) \leftarrow Q_m(D) + \frac{{\sigma_m}^2}{[\min(\mathbf{\Sigma}_t)]^2} \times [\widetilde{Q}_{t,\min}(D) - Q_m(D)] + \mathcal{N}(0,1) \times \sqrt{{\sigma_m}^2 - \frac{{\sigma_m}^4}{[\min(\mathbf{\Sigma}_t)]^2}}$. Note that if Q_m is multidimensional, independent Gaussian noise will be added to each dimension according to the above formula. This also applies to other places of this chapter.

Case 2C): This case considers that σ_m is greater than $\min(\mathbf{\Sigma}_t)$ and σ_m is different from all values in $\mathbf{\Sigma}_t$. Let σ_ℓ be the maximal possible value in $\mathbf{\Sigma}_t$ that is also smaller than σ_m; i.e., $\sigma_\ell = \max\{\sigma_j : \sigma_j \in \mathbf{\Sigma}_t \text{ and } \sigma_j < \sigma_m\}$. Then $\widetilde{Q}_m(D)$ is set as $\widetilde{Q}_\ell(D) + \mathcal{N}(0,1) \times \sqrt{{\sigma_m}^2 - {\sigma_\ell}^2}$. This will become clear by Theorem 1's Result (ii) to be presented in Section 4.1.5.

An example to explain Algorithm 7. Table 4.2 provides an example for better understanding of Algorithm 7. We consider three types of queries. In particular, $Q_1, Q_4, Q_6, Q_{10}, Q_{12}$ are type 1-queries; $Q_2, Q_5, Q_8, Q_9, Q_{11}$ are type 2-queries, and Q_3, Q_7, Q_{13} are type 3-queries.

TABLE 4.2: An example to explain Algorithm 7.

Q_m's query type	Q_1 =type-1	Q_2 =type-2	Q_3 =type-3	Q_4 =type-1	Q_5 =type-2	Q_6 =type-1	Q_7 =type-3
σ_m computed by Line 1 of Alg. 7	$\sigma_1 = 1$	$\sigma_2 = 3$	$\sigma_3 = 2$	$\sigma_4 = 2.5$	$\sigma_5 = 2$	$\sigma_6 = 0.5$	$\sigma_7 = 2$
Case involved in Alg. 7	1): $\widetilde{Q}_1 \leftarrow Q_1$ + $\mathcal{N}(0,1) \times \sigma_1$ with accessing D	1): $\widetilde{Q}_2 \leftarrow Q_2$ + $\mathcal{N}(0,1) \times \sigma_2$ with accessing D	1): $\widetilde{Q}_3 \leftarrow Q_3$ + $\mathcal{N}(0,1) \times \sigma_3$ with accessing D	2C): $\widetilde{Q}_4$ reuses $\widetilde{Q}_1$ without accessing D	2B): $\widetilde{Q}_5$ reuses $\widetilde{Q}_2$ with accessing D	2B): $\widetilde{Q}_6$ reuses $\widetilde{Q}_1$ with accessing D	2A): $\widetilde{Q}_7$ reuses $\widetilde{Q}_3$ without accessing D

Q_8 =type-2	Q_9 =type-2	Q_{10} =type-1	Q_{11} =type-2	Q_{12} =type-1	Q_{13} =type-3
$\sigma_8 = 2.5$	$\sigma_9 = 1.5$	$\sigma_{10} = 0.25$	$\sigma_{11} = 1$	$\sigma_{12} = 0.75$	$\sigma_{13} = 1.5$
2C): $\widetilde{Q}_8$ reuses $\widetilde{Q}_5$ without accessing D	2B): $\widetilde{Q}_9$ reuses $\widetilde{Q}_5$ with accessing D	2B): $\widetilde{Q}_{10}$ reuses $\widetilde{Q}_6$ with accessing D	2B): $\widetilde{Q}_{11}$ reuses $\widetilde{Q}_9$ with accessing D	2C): $\widetilde{Q}_{12}$ reuses $\widetilde{Q}_6$ without accessing D	2B): $\widetilde{Q}_{13}$ reuses $\widetilde{Q}_7$ with accessing D

4.1.5 Explaining the Noise Reuse Rules of Algorithm 7

Our noise-reuse rules of Algorithm 7 are designed to minimize the accumulated privacy costs. To explain this, inspired by [13], we define the privacy loss to quantify privacy costs. We analyze the privacy loss to characterize how privacy degrades in a fine-grained manner, instead of using the composition theorem by Kairouz *et al.* [151]. Although [151] gives the state-of-the-art results for the composition of differentially private algorithms, the results do not assume the underlying mechanisms to achieve differential privacy. In our analysis, by analyzing the privacy loss of Gaussian mechanisms specifically, we can reduce the privacy costs.

When applying a randomized mechanism $\mathcal{M}_G$ to the neighbouring datasets D and D', we use $L_{\mathcal{M}_G}(D, D'; y)$ to represent the privacy loss which denotes the multiplicative difference between the probabilities if the same output y is observed. The definition of the privacy loss is

$$L_{\mathcal{M}_G}(D, D'; y) := \ln \frac{\mathbb{F}\left[\mathcal{M}_G(D) = y\right]}{\mathbb{F}\left[\mathcal{M}_G(D') = y\right]}, \tag{4.1}$$

where $\mathbb{F}\left[\cdot\right]$ represents the probability density function, and y represents the output.

In Eq. (4.1), the probability density function $\mathbb{F}\left[\cdot\right]$ is used for simplicity when assuming the randomized mechanism $\mathcal{M}_G$ has a continuous output. If the output of $\mathcal{M}_G$ is discrete, probability mass function $\mathbb{P}\left[\cdot\right]$ is used to replace $\mathbb{F}\left[\cdot\right]$.

When $\mathcal{M}_G(D)$ is the probability distribution of a random variable y, $L_{\mathcal{M}_G}(D, D'; \mathcal{M}_G(D))$ is the probability distribution of $L_{\mathcal{M}_G}(D, D'; y)$. We simplify $L_{\mathcal{M}_G}(D, D'; \mathcal{M}_G(D))$ as $L_{\mathcal{M}_G}(D, D')$.

We denote the composition of some randomized mechanisms $\mathcal{M}_{G1}, \mathcal{M}_{G2}, \ldots, \mathcal{M}_{Gm}$ for a positive integer m by $\mathcal{M}_{G1} \| \mathcal{M}_{G2} \| \ldots \| \mathcal{M}_{Gm}$. For the composition, the privacy loss with respect to neighboring datasets D and D' when the outputs of randomized mechanisms $\mathcal{M}_{G1}, \mathcal{M}_{G2}, \ldots, \mathcal{M}_{Gm}$ are $y_1, y_2, \ldots, y_m$ is defined by

$$L_{\mathcal{M}_{G1} \| \mathcal{M}_{G2} \| \ldots \| \mathcal{M}_{Gm}}(D, D'; y_1, y_2, \ldots, y_m) := \ln \frac{\mathbb{F}\left[\cap_{i=1}^{m} \left[\mathcal{M}_{Gi}(D) = y_i\right]\right]}{\mathbb{F}\left[\cap_{i=1}^{m} \left[\mathcal{M}_{Gi}(D') = y_i\right]\right]}.$$

When y_i follows the probability distribution of random variable $\mathcal{M}_{Gi}(D)$ for each $i \in \{1, 2, \ldots, m\}$, clearly $L_{\mathcal{M}_{G1}\|\mathcal{M}_{G2}\|\ldots\|\mathcal{M}_{Gm}}(D, D'; y_1, y_2, \ldots, y_m)$ follows the probability distribution of random variable

$$L_{\mathcal{M}_{G1}\|\mathcal{M}_{G2}\|\ldots\|\mathcal{M}_{Gm}}(D, D'; \mathcal{M}_{G1}(D), \mathcal{M}_{G2}(D), \ldots, \mathcal{M}_{Gm}(D)),$$

which we write as $L_{\mathcal{M}_{G1}\|\mathcal{M}_{G2}\|\ldots\|\mathcal{M}_{Gm}}(D, D')$ for simplicity.

With the privacy loss defined above, we now analyze how to reuse noise when a series of queries are answered under differential privacy. To this end, we present Theorem 1, which presents the optimal ratio of reusing noise to minimize privacy costs.

Theorem 1 (**Optimal ratio of reusing noise to minimize privacy costs**). Suppose that before answering query Q_m and after answering $Q_1, Q_2, \ldots, Q_{m-1}$, the privacy loss $L_{\widetilde{Q}_1\|\widetilde{Q}_2\|\ldots\|\widetilde{Q}_{m-1}}(D, D')$ is given by $\mathcal{N}(\frac{\mathcal{A}(D,D')}{2}, \mathcal{A}(D, D'))$ for some $\mathcal{A}(D, D')$. For the m-th query Q_m, suppose that Q_m is the same as Q_j for some $j \in \{1, 2, \ldots, m-1\}$ and we reuse r fraction of noise in $\widetilde{Q}_j(D)$ to generate $\widetilde{Q}_m(D)$ for $0 \leq r \leq 1$ satisfying ${\sigma_m}^2 - r^2{\sigma_j}^2 > 0$, where r is a constant to be decided. If $\widetilde{Q}_j(D) - Q_j(D)$ follows a Gaussian probability distribution with mean 0 and standard deviation σ_j, we generate the noisy response $\widetilde{Q}_m(D)$ to answer query Q_m as follows:

$$\widetilde{Q}_m(D) \leftarrow Q_m(D) + r[\widetilde{Q}_j(D) - Q_j(D)] + \mathcal{N}(0, {\sigma_m}^2 - r^2{\sigma_j}^2), \tag{4.2}$$

so that $\widetilde{Q}_m(D) - Q_m(D)$ follows a Gaussian probability distribution with mean 0 and standard deviation σ_m.

Note that Δ_{Q_m} and Δ_{Q_j} are the same since Q_m and Q_j are the same. Then we have the following results.

(i) After answering the m queries $Q_1, Q_2, \ldots, Q_m$, the privacy loss $L_{\widetilde{Q}_1\|\widetilde{Q}_2\|\ldots\|\widetilde{Q}_m}(D, D')$ will be $\mathcal{N}(\frac{B_r(D,D')}{2}, B_r(D, D'))$ for $B_r(D, D') := \mathcal{A}(D, D') + \frac{[\|Q_m(D) - Q_m(D')\|_2]^2(1-r)^2}{{\sigma_m}^2 - r^2{\sigma_j}^2}$.

(ii) We clearly require $r \geq 0$ and ${\sigma_m}^2 - r^2{\sigma_j}^2 \geq 0$ in (4.2) above (note that $\mathcal{N}(0, 0) \equiv 0$). To minimize the total privacy costs (which is equivalent to

minimize $B_r(D, D')$ above), the optimal r is given by

$$r_{\text{optimal}} = \begin{cases} 1, & \text{if } \sigma_m \geq \sigma_j, \\ \left(\frac{\sigma_m}{\sigma_j}\right)^2, & \text{if } \sigma_m < \sigma_j, \end{cases} \tag{4.3}$$

so that substituting Eq. (4.3) into the expression of $B_r(D, D')$ gives

$$\begin{aligned} &B_{r_{\text{optimal}}}(D, D') \\ &= \begin{cases} \mathcal{A}(D, D'), \text{ if } \sigma_m \geq \sigma_j; \\ \mathcal{A}(D, D') + [\|Q_m(D) - Q_m(D')\|_2]^2 \left(\frac{1}{{\sigma_m}^2} - \frac{1}{{\sigma_j}^2}\right), \text{if } \sigma_m < \sigma_j. \end{cases} \end{aligned} \tag{4.4}$$

Note that if $\sigma_m = \sigma_j$ for some $j \in \{1, 2, \ldots, m-1\}$, we have $r_{\text{optimal}} = 1$ and just set $\widetilde{Q}_m(D)$ as $\widetilde{Q}_j(D)$.

Proof. The proof is in Appendix B.1. □

Eq. (4.3) of Theorem 1 clearly indicates the noise use ratio $\frac{{\sigma_m}^2}{[\min(\boldsymbol{\Sigma}_t)]^2}$ of Case 2B) in Algorithm 7 (see Line 22 of Algorithm 7), and the noise use ratio 1 of Cases 2A) and 2C) in Algorithm 7 (see Lines 15 and 30 of Algorithm 7).

By considering $r = 0$ in Result (i) of Theorem 1, we obtain Corollary 1, which presents the classical result on the privacy loss of a single run of the Gaussian mechanism.

Corollary 1. By considering $m = 1$ in Result (i) of Theorem 1, we have that for a randomized algorithm $\widetilde{Q}$ which adds Gaussian noise amount σ to a query Q, the privacy loss with respect to neighboring datasets D and D' is given by $\mathcal{N}(\frac{\mathcal{A}(D,D')}{2}, \mathcal{A}(D, D'))$ for $\mathcal{A}(D, D') := \frac{[\|Q(D)-Q(D')\|_2]^2}{\sigma^2}$.

Corollary 1 has been shown in many prior studies [8–10] on the Gaussian mechanism for differential privacy.

By considering $r = 0$ in Result (i) of Theorem 1, we obtain Corollary 2, which presents the privacy loss of the naive algorithm where the noisy response to each query is generated independently using fresh noise.

Corollary 2 (**Privacy loss of the naive algorithm where each query is answered independently**). Suppose a dataset has been used to answer n queries

$Q_1, Q_2, \ldots, Q_n$ under differential privacy. Specifically, for $i = 1, 2, \ldots, n$, to answer the i-th query Q_i under (ϵ_i, δ_i)-differential privacy, a noisy response $\widetilde{Q}_i$ is generated by adding independent Gaussian noise $\sigma_i := \text{Gaussian}(\Delta_{Q_i}, \epsilon_i, \delta_i)$ to the true query result Q_i, where Δ_{Q_i} is the ℓ_2-sensitivity of Q_i. Then after answering n queries $Q_1, Q_2, \ldots, Q_n$ independently as above, the privacy loss with respect to neighboring datasets D and D' is given by $\mathcal{N}(\frac{F(D,D')}{2}, F(D, D'))$ for $F(D, D') := \sum_{i=1}^{n} \frac{[\|Q_i(D) - Q_i(D')\|_2]^2}{{\sigma_i}^2}$.

4.1.6 Explaining Privacy Cost Update in Algorithm 7

Among the above cases, Cases 2A) and 2C) do not incur additional privacy costs since they just use previous noisy results and generate fresh Gaussian noise, without access to the dataset D. In contrast, Cases 1) and 2B) incur additional privacy costs since they need to access the dataset D to compute the true query result $Q_m(D)$. Hence, in Algorithm 7, the privacy cost is updated in Cases 1) and 2B), but not in Cases 2A) and 2C). In this section, we explain the reason that the privacy cost is updated in Algorithm 7 according to Lines 3 and 5 for Case 1), and Lines 18 and 19 for Case 2B).

When our Algorithm 7 is used, we let the above randomized mechanism $\mathcal{M}_{Gi}$ be our noisy response function $\widetilde{Q}_i$. When $\widetilde{Q}_1, \widetilde{Q}_2, \ldots, \widetilde{Q}_{i-1}$ on dataset D are instantiated as $y_1, y_2, \ldots, y_{i-1}$, if the generation of $\widetilde{Q}_i$ on dataset D uses $\widetilde{Q}_j$ for some $j < i$, then the auxiliary information aux_i in the input to $\widetilde{Q}_i$ contains y_j (aux_1 is $\emptyset$). For the consecutive use of our Algorithm 7, it will become clear that the privacy loss, defined by $L_{\widetilde{Q}_1, \widetilde{Q}_2, \ldots, \widetilde{Q}_m}(y_1, y_2, \ldots, y_m) := \ln \max_{\text{neighboring datasets } D, D'} \frac{\mathbb{F}\left[\cap_{i=1}^{m} \left[\widetilde{Q}_i(D) = y_i\right]\right]}{\mathbb{F}\left[\cap_{i=1}^{m} \left[\widetilde{Q}_i(D') = y_i\right]\right]}$, follows a Gaussian probability distribution with mean $\frac{V}{2}$ and variance V for some V, denoted by $\mathcal{N}(\frac{V}{2}, V)$. For such a reason that form of privacy loss, the corresponding differential-privacy level is given by the following lemma.

Lemma 14. If the privacy loss of a randomized mechanism $\mathcal{M}_G$ with respect to neighboring datasets D and D' is given by $\mathcal{N}(\frac{V(D,D')}{2}, V(D, D'))$ for some $V(D, D')$, then $\mathcal{M}_G$ achieves (ϵ, δ)-differential privacy for ϵ and δ satisfying $\max_{\text{neighboring datasets } D, D'} V(D, D') = [\text{Gaussian}(1, \epsilon, \delta)]^{-2}$.

Proof. The proof details are in Appendix B.2. □

Based on the privacy loss defined above, we have the following theorem which explains the rules to update the privacy costs in our Algorithm 7.

Theorem 2. We consider the consecutive use of Algorithm 7 here. Suppose that after answering $Q_1, Q_2, \ldots, Q_{m-1}$ and before answering query Q_m, the privacy loss with respect to neighboring datasets D and D' is given by $\mathcal{N}(\frac{\mathcal{A}(D,D')}{2}, \mathcal{A}(D,D'))$ for some $\mathcal{A}(D,D')$, and the corresponding privacy level can be given by $(\epsilon_{\text{old}}, \delta_{\text{budget}})$-differential privacy. Then, in Algorithm 7, after answering all m queries $Q_1, Q_2, \ldots, Q_{m-1}, Q_m$, we have:

- the privacy loss with respect to neighboring datasets D and D'
  - ① will still be $\mathcal{N}(\frac{\mathcal{A}(D,D')}{2}, \mathcal{A}(D,D'))$ in Cases 2A) and 2C),
  - ② will be $\mathcal{N}(\frac{B(D,D')}{2}, B(D,D'))$ in Case 1) for $B(D,D') := \mathcal{A}(D,D') + \frac{[\|Q_m(D)-Q_m(D')\|_2]^2}{{\sigma_m}^2}$,
  - ③ will be $\mathcal{N}(\frac{C(D,D')}{2}, C(D,D'))$ in Case 2B) for $C(D,D') := \mathcal{A}(D,D') + [\|Q_m(D) - Q_m(D')\|_2]^2 \times \left[\frac{1}{{\sigma_m}^2} - \frac{1}{[\min(\mathbf{\Sigma}_t)]^2}\right]$;
- the corresponding privacy level can be given by $(\epsilon_{\text{new}}, \delta_{\text{budget}})$-differential privacy with the following ϵ_{new}:
  - ④ $\epsilon_{\text{new}} = \epsilon_{\text{old}}$ in Cases 2A) and 2C),
  - ⑤ ${\epsilon_{\text{new}}}^2 = {\epsilon_{\text{old}}}^2 + \epsilon\_\text{squared\_cost}$ in Case 1) for $\epsilon\_\text{squared\_cost}$ satisfying $\text{Gaussian}(\Delta_{Q_m}, \sqrt{\epsilon\_\text{squared\_cost}}, \delta_{\text{budget}}) = \sigma_m$,
  - ⑥ ${\epsilon_{\text{new}}}^2 = {\epsilon_{\text{old}}}^2 + \epsilon\_\text{squared\_cost}$ in Case 2B) for $\epsilon\_\text{squared\_cost}$ satisfying $[\text{Gaussian}(\Delta_{Q_m}, \sqrt{\epsilon\_\text{squared\_cost}}, \delta_{\text{budget}})]^{-2} = {\sigma_m}^{-2} - [\min(\mathbf{\Sigma}_t)]^{-2}$.

Theorem 2 explains the rules to update the privacy cost in Algorithm 7. Specifically, Result ⑤ gives Lines 3 and 5 for Case 1), and Result ⑥ gives Lines 18 and 19 for Case 2B).

Proof. The proof is in Appendix B.3. □

4.1.7 Analyzing the Total Privacy Costs

Based on Theorem 2, we now analyze the total privacy costs when our system calls Algorithm 7 consecutively.

At the beginning when no query has been answered, we have $V = 0$ (note that $\mathcal{N}(0,0) \equiv 0$). Then by induction via Corollary 1 and Theorem 2, for the consecutive use of Algorithm 7, the privacy loss is always in the form of $\mathcal{N}(\frac{V}{2}, V)$ for some V. In our Algorithm 7, the privacy loss changes only when the query being answered belongs to Cases 1) and 2B). More formally, we have the following theorem.

Theorem 3. Among queries $Q_1, Q_2, \cdots, Q_n$, let N_1, N_{2A}, N_{2B}, and N_{2C} be the set of $i \in \{1, 2, \ldots, n\}$ such that Q_i is in Cases 1), 2A), 2B), and 2C), respectively. For queries in Case 2B), let T_{2B} be the set of query types. In Case 2B), for query type $t \in T_{2B}$, suppose the number of type-t queries be m_t, and let these type-t queries be $Q_{j_{t,1}}, Q_{j_{t,2}}, \ldots, Q_{j_{t,m_t}}$ for indices $j_{t,1}, j_{t,2}, \ldots, j_{t,m_t}$ (ordered chronologically) all belonging to N_{2B}. From Case 2B) of Algorithm 7, we have $\sigma_{j_{t,1}} > \sigma_{j_{t,2}} > \ldots > \sigma_{j_{t,m_t}}$, and for $k \in \{2, 3, \ldots, m_t\}$, $\widetilde{Q}_{j_{t,k}}$ is answered by reusing $\frac{{\sigma_{j_{t,k}}}^2}{{\sigma_{j_{t,k-1}}}^2}$ fraction of old noise in $\widetilde{Q}_{j_{t,k-1}}$; more specifically, $\widetilde{Q}_{j_{t,k}} = Q_{j_{t,k}} + \frac{{\sigma_{j_{t,k}}}^2}{{\sigma_{j_{t,k-1}}}^2}[\widetilde{Q}_{j_{t,k-1}} - Q_{j_{t,k-1}}] + \mathcal{N}(0, {\sigma_{j_{t,k}}}^2 - \frac{{\sigma_{j_{t,k}}}^4}{{\sigma_{j_{t,k-1}}}^2})$ from Line 22 of Algorithm 7 in Section 4.1.5 for Case 2B). We also consider that $\widetilde{Q}_{j_{t,1}}$ is answered by reusing $\frac{{\sigma_{j_{t,1}}}^2}{{\sigma_{j_{t,0}}}^2}$ fraction of old noise in $\widetilde{Q}_{j_{t,0}}$. Let the ℓ_2-sensitivity of a type-t query be $\Delta(\text{type-}t)$.

In the example provided in Table 4.2, we have $N_1 = \{1, 2, 3\}$, $N_{2A} = \{7\}$, $N_{2B} = \{5, 6, 9, 10, 11, 13\}$, and $N_{2C} = \{8, 12\}$. $T_{2B} = \{\text{type-1}, \text{type-2}, \text{type-3}\}$. In Case 2B), the number of type-1 queries is $m_1 = 2$, and these type-1 queries are Q_6 and Q_{10} so $j_{1,1} = 6$ and $j_{1,2} = 10$ (also $j_{1,0} = 1$ since $\widetilde{Q}_6$ reuses $\widetilde{Q}_1$); the number of type-2 queries is $m_2 = 3$, and these type-2 queries are Q_5, Q_9, and Q_{11} so $j_{2,1} = 5$ and $j_{2,2} = 9$, $j_{2,3} = 11$ (also $j_{2,0} = 2$ since $\widetilde{Q}_5$ reuses $\widetilde{Q}_2$); the number of type-3 queries is $m_3 = 1$, and this type-3 query is Q_{13} so $j_{3,1} = 13$ (also $j_{3,0} = 3$ since $\widetilde{Q}_{13}$ reuses $\widetilde{Q}_3$).

Then after Algorithm 7 is used to answer all n queries with query Q_i being answered under (ϵ_i, δ_i)-differential privacy, we have:

- The total privacy loss with respect to neighboring datasets D and D' is given by $\mathcal{N}(\frac{G(D,D')}{2}, G(D,D'))$, where

$$G(D,D') := \sum_{i\in N_1} \frac{[\|Q_i(D) - Q_i(D')\|_2]^2}{{\sigma_i}^2} + \sum_{t\in T_{2B}} \left\{ \frac{[\|Q_{j_{t,m_t}}(D) - Q_{j_{t,m_t}}(D')\|_2]^2}{{\sigma_{j_{t,m_t}}}^2} - \frac{[\|Q_{j_{t,0}}(D) - Q_{j_{t,0}}(D')\|_2]^2}{{\sigma_{j_{t,0}}}^2} \right\}, \tag{4.5}$$

and the first summation is the contribution from queries in Case 1), and the second summation is the contribution from queries in Case 2B). When D and D' iterate the space of neighboring datasets, the maximum of $\|Q_i(D) - Q_i(D')\|$ is Q_i's ℓ_2-sensitivity Δ_{Q_i}, and the maximum of both $\|Q_{j_{t,m_t}}(D) - Q_{j_{t,m_t}}(D')\|_2$ and $\|Q_{j_{t,0}}(D) - Q_{j_{t,0}}(D')\|_2$ are $\Delta(\text{type-}t)$ since $Q_{j_{t,m_t}}$ and $Q_{j_{t,0}}$ are both type-t queries, we obtain

$$\begin{aligned} &\max_{\text{neighboring datasets } D,\, D'} G(D,D') \\ &= \sum_{i\in N_1} \frac{{\Delta_{Q_i}}^2}{{\sigma_i}^2} + \sum_{t\in T_{2B}} \left[\frac{[\Delta(\text{type-}t)]^2}{{\sigma_{j_{t,m_t}}}^2} - \frac{[\Delta(\text{type-}t)]^2}{{\sigma_{j_{t,0}}}^2} \right]. \end{aligned} \tag{4.6}$$

In the example provided in Table 4.2 in Section 4.2, $\max_{\text{neighboring datasets } D,\, D'} G(D,D')$ is given by

$$\begin{aligned} &\frac{{\Delta_{Q_1}}^2}{{\sigma_1}^2} + \frac{{\Delta_{Q_2}}^2}{{\sigma_2}^2} + \frac{{\Delta_{Q_3}}^2}{{\sigma_3}^2} + \left[\frac{[\Delta(\text{type-1})]^2}{{\sigma_{10}}^2} - \frac{[\Delta(\text{type-1})]^2}{{\sigma_1}^2} \right] \\ &+ \left[\frac{[\Delta(\text{type-2})]^2}{{\sigma_{11}}^2} - \frac{[\Delta(\text{type-2})]^2}{{\sigma_2}^2} \right] + \left[\frac{[\Delta(\text{type-3})]^2}{{\sigma_{13}}^2} - \frac{[\Delta(\text{type-3})]^2}{{\sigma_3}^2} \right] \\ &= \frac{[\Delta(\text{type-1})]^2}{{\sigma_{10}}^2} + \frac{[\Delta(\text{type-2})]^2}{{\sigma_{11}}^2} + \frac{[\Delta(\text{type-3})]^2}{{\sigma_{13}}^2}. \end{aligned}$$

- From Lemma 14, the total privacy costs of our Algorithm 7 can be given by $(\epsilon_{\text{ours}}, \delta_{\text{budget}})$-differential privacy for ϵ_{ours} satisfying

$$[\text{Gaussian}(1, \epsilon_{\text{ours}}, \delta_{\text{budget}})]^{-2} = \max_{\text{neighboring datasets } D,\, D'} G(D,D'), \tag{4.7}$$

or (ϵ, δ)-differential privacy for any ϵ and δ satisfying $[\text{Gaussian}(1, \epsilon, \delta)]^{-2} = \max_{\text{neighboring datasets } D, D'} G(D, D')$.

Proof. The proof is in Appendix B.4. □

Remark 4.1. Theorem 3 can be used to understand that our Algorithm 7 incurs fewer privacy costs than that of the naive algorithm where n queries are answered independently. As given in Corollary 2, the privacy loss with respect to neighboring datasets D and D' is given by $\mathcal{N}(\frac{F(D,D')}{2}, F(D, D'))$ for $F(D, D') := \sum_{i=1}^{n} \frac{[\|Q_i(D) - Q_i(D')\|_2]^2}{{\sigma_i}^2}$. Clearly, $F(D, D') \geq G(D, D')$ for $G(D, D')$ given by Eq. (4.5) above. From Lemma 14, the privacy cost of the naive algorithm can be given by $(\epsilon_{\text{naive}}, \delta_{\text{budget}})$-differential privacy for ϵ_{naive} satisfying $[\text{Gaussian}(1, \epsilon_{\text{naive}}, \delta_{\text{budget}})]^{-2} = \max_{\text{neighboring datasets } D, D'} F(D, D')$, which with Eq. (4.7) in Theorem 3 and the expression of $\text{Gaussian}(\cdot, \cdot, \cdot)$ in Lemma 2.1 implies

$$\frac{\epsilon_{\text{ours}}}{\epsilon_{\text{naive}}} = \sqrt{\frac{\max_{\text{neighboring datasets } D, D'} G(D, D')}{\max_{\text{neighboring datasets } D, D'} F(D, D')}} \leq 1,$$

where the equal sign is taken only when all n queries are different so no noise reuse is incurred in our Algorithm 7.

4.1.8 Computing the ℓ_2-sensitivity of A Query

For one-dimensional real-valued query Q, Δ_Q is simply the maximal absolute difference between $Q(D)$ and $Q(D')$ for any neighboring datasets D and D'. In Section 4.3, to evaluate the performance, we define neighboring datasets by considering modifying an entry. Then, if the dataset has n users' information, and the domain of each user's income is within the interval $[\text{min\_income}, \text{max\_income}]$, then Δ_Q for query Q being the average income of all users is $\frac{\text{max\_income} - \text{min\_income}}{n}$ since this is the maximal variation in the output when a user's record changes. Similarly, Δ_Q for query Q being the percentage of female users is $\frac{1}{n}$.

4.2 Implementation Challenges of Our Blockchain-Based System

We now discuss challenges and countermeasures during the design and implementation of our blockchain-based system.

Smart Contract fetches external data. Ethereum blockchain applications, such as Bitcoin scripts and smart contracts are unable to access and fetch directly the external data they need. However, in our application, `Blockchain` needs to fetch data from `Server` then returns them to `Client`. This requires smart contract to send the HTTP POST request. Hence, we use the Provable, a service integrated with a number of blockchain protocols and can be accessed by non-blockchain applications as well. It guarantees that data fetched from the original data-source is genuine and untampered.

By using the Provable, smart contracts can directly access data from web sites or APIs. In our case, `Blockchain` can send HTTP requests to `Server` with parameters, and then process and store data after `Server` responds successfully.

Mathematical operations with Solidity. `Blockchain` is written using solidity language which is designed to target Ethereum Virtual Machine. However, current solidity language does not have inherent functions for complex mathematical operations, such as taking the square root or logarithm. We write a function to implement the square root operation. To avoid using Lemma 2.1 to compute logarithm in `Blockchain`, we generate Gaussian noise in `Client`, and pass the value to `Blockchain` as one of the parameters in function QueryMatch. Besides, current Solidity version cannot operate float or double type data. To keep the precision, we scale up the noise amount during calculation, and then scale down the value before returning the noisy data to analysts.

4.3 Implementation and Experiments

In this section, we perform experiments to validate that the proposed system and algorithm are effective in saving privacy costs according to the system flow shown in Fig. 4.2. More specifically, a user sends a query through the UI, and then `Client`

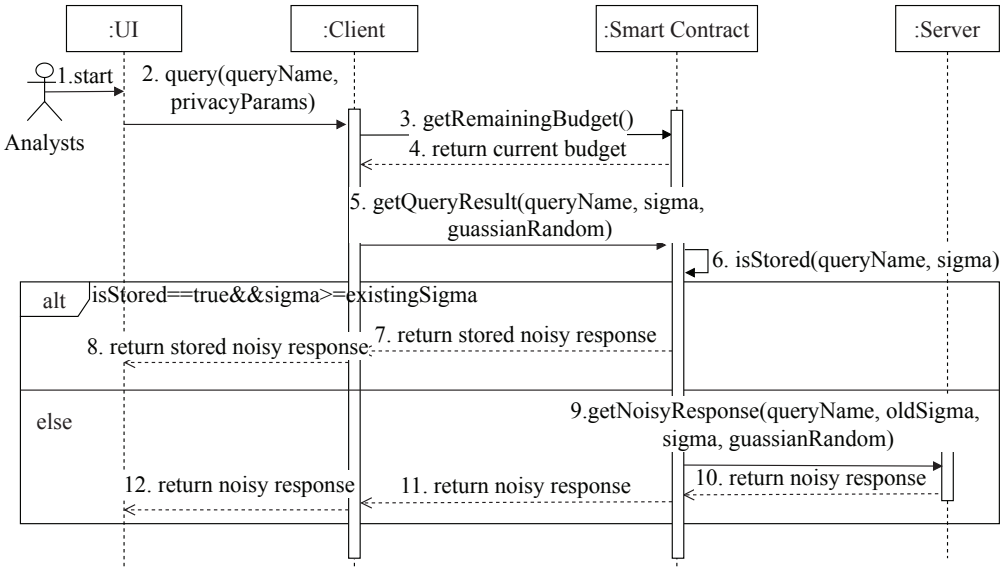


FIGURE 4.2: The proposed blockchain-based system working flow for differential-privacy costs management.

receives the query and forwards it to `Blockchain` smart contract. After the smart contract checks with stored data, it will decide whether to return the noisy response to `Client` directly or forward the request to `Server`. If `Server` receives the request, it will generate and return a noisy response to the smart contract.

4.3.1 Experiment Setup

We prototype a web application based on the system description in Section 4.1. We use the Javascript language to write `Client`, whereas the Solidity language is for `Blockchain` smart contract. Besides, Web3 is used as the Javascript API to exchange information between `Client` and `Blockchain` smart contract, and then Node.js and Express web framework are leveraged to set up `Server`. In addition, MongoDB is used as the database to host the real-world dataset. Our designed smart contracts are deployed on the Ropsten [152] testnet with the MetaMask extension of the Chrome browser. The Ropsten testnet is a testing blockchain environment maintained by Ethereum, and it implements the same Proof-of-Work protocol as the main Ethereum network. Fig. 4.3 shows the screenshot of our blockchain-based privacy management system. Fig. 4.4 presents outputs while sending queries using the system.

The evaluation of the proposed differential privacy mechanism is based on a real-world dataset containing American community survey samples extracted from the *Integrated Public Use Microdata Series* at `https://www.ipums.org`. There are 5000 records in the dataset. Each record includes the following numerical attributes: "Total personal income", "Total family income", "Age", and categorical attributes: "Race", "Citizenship status". We set the privacy budget as $\epsilon_{\text{budget}} = 8$ and $\delta_{\text{budget}} = 10^{-4}$, which are commonly used to protect the privacy of a dataset [18, 153]. We consider five types of queries: "average personal income", "average total family income", "frequency of US citizens", "frequency of white race", and "frequency of age more than 60". For the privacy parameter of each query Q_i, we sample ϵ_i uniformly from $[0.1, 1.1]$ and sample δ_i uniformly from $[10^{-7}, 10^{-5}]$. The sensitivities of these queries are 202, 404, 0.0002, 0.0002, and 0.0002, respectively. We compute the sensitivity of a query based on Section 4.1.8. For the query "average total personal income", since the user's total personal income ranges from -5000 to 700000 in the dataset mentioned above, we assume the domain of total personal income is in the range of $[-10000, 1000000]$ for all possible datasets. The sensitivity is $(1000000 - (-10000))/5000 = 202$ and the mechanism protects the privacy of all data within $[-10000, 1000000]$. Thus, it can protect the privacy of the dataset in our experiment. Suppose the received query is "average total family income". In that case, we assume the maximal variation is $[-20000, 2000000]$ for all possible datasets because the total family income's range is $[-5000, 1379500]$ in the dataset we use. The sensitivity is $(2000000 - (-20000))/5000 = 404$. Hence, our generated noise with the sensitivity of 404 can protect the privacy of all data within $[-20000, 2000000]$. Therefore, it can protect the privacy of the dataset we use as well. The sensitivity for queries "frequency of US citizens", "frequency of white race", and "frequency of age more than 60" is $1/5000 = 0.0002$.

4.3.2 Experimental Results

The benchmark of our experiment is a naive scheme which does not contain Algorithm 7 in the smart contract. That is, every query will be forwarded by the smart contract to `Server` to get the noisy response. Hence, no differential privacy cost can be reused in the naive scheme.

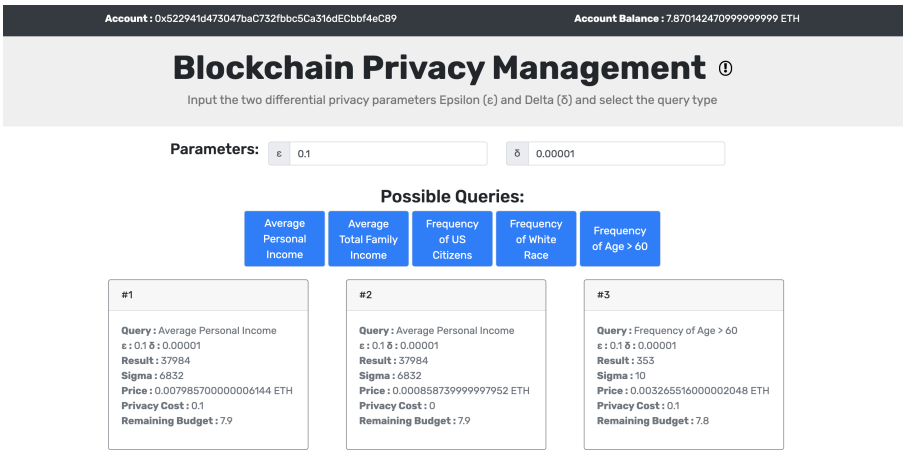


FIGURE 4.3: Screenshot of blockchain-based privacy management system demo.

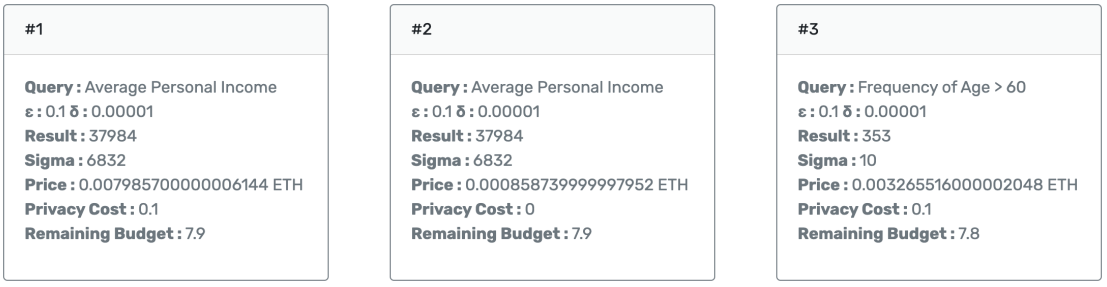


FIGURE 4.4: Displaying of outputs with ϵ privacy costs.

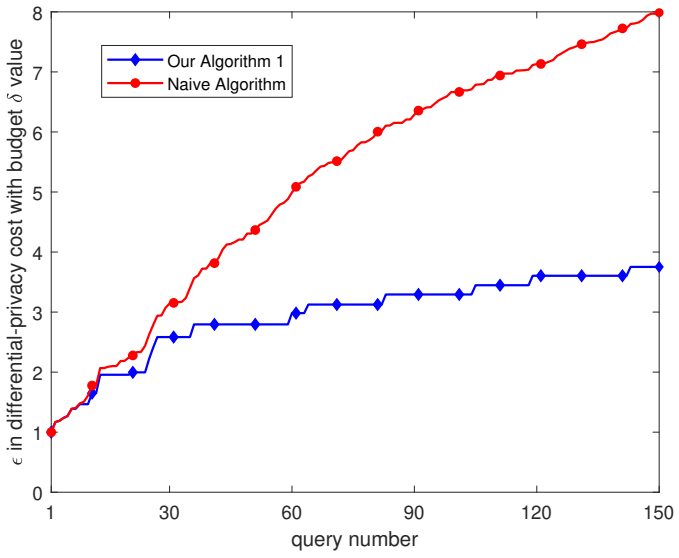


FIGURE 4.5: Performance comparison of the sum of privacy costs.

First, we use an experiment to validate that our proposed Algorithm 7 is effective in saving privacy costs. Thus, we design a performance comparison experiment by

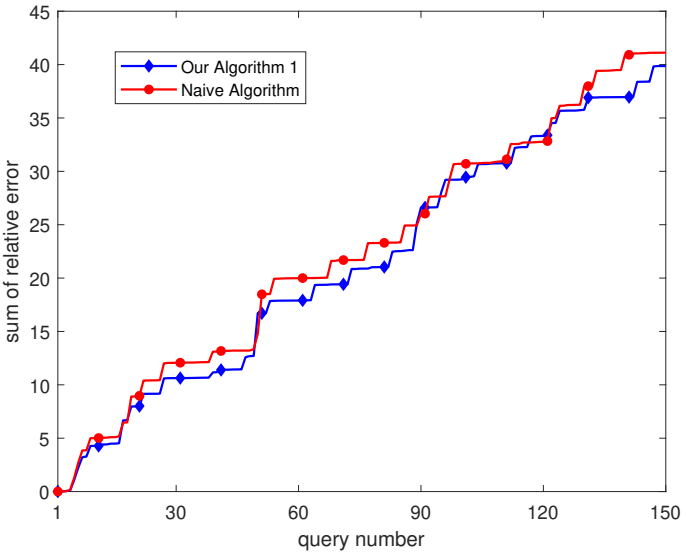


FIGURE 4.6: Performance comparison of the sum of relative error.

tracking privacy costs using our Algorithm 7 and the naive scheme, respectively. Specifically, we deploy two smart contracts implementing our Algorithm 7 and the naive scheme respectively on the Ropsten testnet. Then, we send 150 requests randomly selected in five query types from `Client` of the web application, and record the privacy cost of each query. As shown in Fig. 4.5, compared with the naive scheme, the proposed algorithm saves significant privacy cost. When the number of the queries is 150, the differential-privacy cost of Algorithm 7 is about 52% less than that of the naive algorithm. We also observe that the privacy cost in the proposed scheme increases slowly when the number of queries increases, even trending to converge to a specific value. The reason is that, in Algorithm 7, for each query type, we can always partially or fully reuse previous noisy answers when the query type is asked for a second time or more. Therefore, in our scheme, many queries are answered without incurring the additional privacy cost if noisy responses fully reuse previous noisy answers.

Second, to prove that the proposed Algorithm 7 retains the accuracy of the dataset, we design another experiment to compare the sum of relative errors. We use the same smart contracts as those in the last experiment. We accumulate relative errors incurred in each query. Fig. 4.6 shows that the sum of relative errors of Algorithm 7 is comparable with that of the naive scheme. Since relative errors are similar between two schemes, our results demonstrate that the proposed Algorithm 7 keeps the accuracy.

As a summary, Fig. 4.5 and Fig. 4.6 together demonstrate that our Algorithm 7 can save privacy costs significantly without sacrificing the accuracy of the dataset.

Third, we evaluate the relationship between the query utility and the privacy budget. As defined in [154], the privacy utility of a mechanism satisfies (α, β)-useful if $|\widetilde{Q}_m(D) - Q_m(D)| \leq \alpha$ with probability at least $1 - \beta$. Thus, a small α means that the gap between the perturbed result and the actual result is small, which also reflects that the mechanism has a high utility. The noise added to a query can be calculated as $\sigma = \text{Gaussian}(\Delta_Q, \epsilon, \delta)$, where $\text{Gaussian}(\Delta_Q, \epsilon, \delta) = \sqrt{2 \ln \frac{1.25}{\delta}} \times \frac{\Delta_Q}{\epsilon}$. We set $\delta = 10^{-5}$ and $\epsilon \in [1, 8]$. Appendix B.5 proves that when we set $\beta = 0.05$, $\alpha = 2\sigma$. Fig. 4.7 and Fig. 4.8 illustrate how the utility and noise change as the privacy budget ϵ increases. Fig. 4.7 shows the value of α decreases when the privacy budget ϵ increases, meaning that the utility increases. In addition, the amount of noise added reflects the utility of the query as well. When less noise is added to the query response, the more utility the response gains. Fig. 4.8 shows that how the noise changes with the privacy budget. As the privacy budget increases, noise decreases, which means that the query utility increases. The amount of noise depends on values of the privacy budget and the sensitivity. Queries such as "Frequency of US citizens", "Frequency of white race" and "Frequency of age more than 60" have the same sensitivity value 0.0002, so the noise added to their responses is the same when the privacy budgets they use are equal.

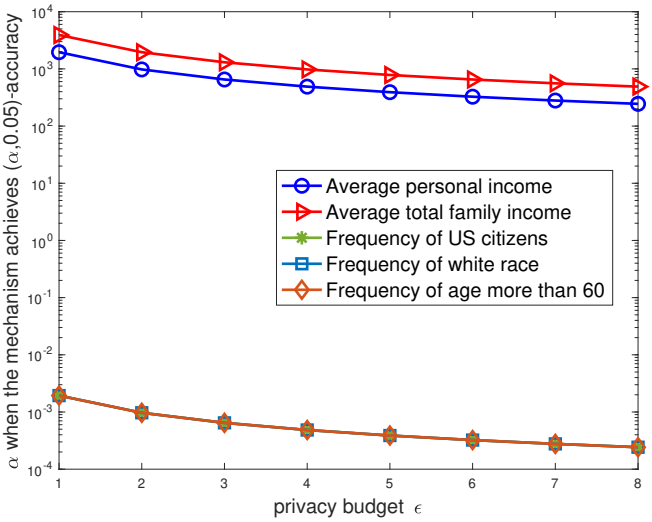


FIGURE 4.7: Utility vs the privacy budget.

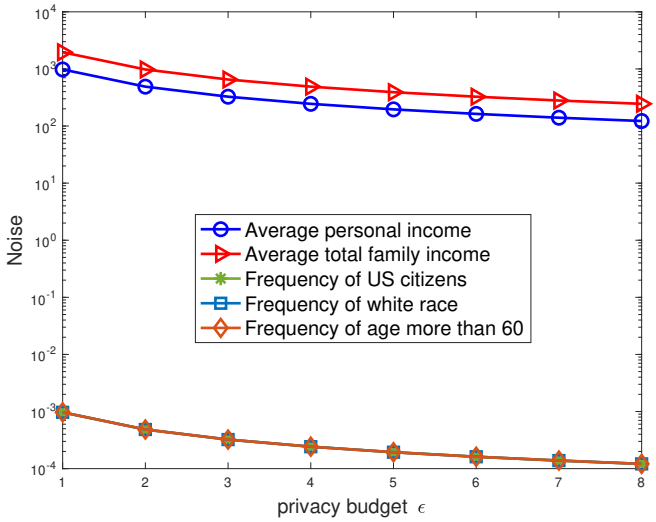


FIGURE 4.8: Noise vs the privacy budget.

4.4 Summary

In this chapter, we use a blockchain-based approach for tracking and saving differential privacy costs. Our design proposes an algorithm that reuses noise fully or partially for different instances of the same query type to minimize the accumulated privacy costs. The efficiency of the algorithm is proved via rigorous mathematical proof. Moreover, we design a blockchain-based system for conducting real-world experiments to confirm the proposed approach is effective.

Chapter 5

Privacy-Preserving Blockchain-Based Federated Learning for IoT Devices[1]

In this chapter, we design an FL-based system that considers home appliances of the same brand in a family as a unit, and a mobile phone is used to collect data from home appliances periodically and train the machine learning model locally [44]. Besides, our designed system uses blockchain to record crowdsourcing activities. Moreover, we improve the traditional batch normalization by removing constraints of mean value and variance while constraining the bound within $[-\sqrt{N-1}, \sqrt{N-1}]$, where N denotes the batch size. Then, we add the Laplace noise to the normalization layer with a formal privacy guarantee to perturb the extracted features so that the trained models can prevent information leakage. Furthermore, to motivate more customers to participate in the crowdsourcing task and reduce malicious and poisoning updates, we utilize a reputation-based crowdsourcing incentive mechanism, which rewards reliable customers and punishes malicious customers correspondingly. We evaluate the FL algorithm with Convolutional Neural Networks (CNN) for MNIST digit recognition task. A global model is initiated with random parameters, and then each participant trains the model using their data. All the locally trained models are aggregated to obtain a set of averaged

[1]The work in this chapter has been published as Yang Zhao, Jun Zhao, Linshan Jiang, Rui Tan, Dusit Niyato, Zengxiang Li, Lingjuan Lyu, and Yingbo Liu. "Privacy-Preserving Blockchain-Based Federated Learning for IoT Devices", in *IEEE Internet of Things Journal, DOI: 10.1109/JIOT.2020.3017377, 2021.*

model parameters. During the training, the batch normalization technique is commonly adopted in order to resolve the overfitting issue.

Contributions. The major contributions of this chapter are summarized as follows:

- First, a hierarchical crowdsourcing FL system is proposed to build the machine learning model to help home appliance manufacturers improve their service quality and optimize functionalities of home appliances.
- Second, we propose a new normalization technique that delivers a higher test accuracy than batch normalization while preserving the privacy of the extracted features of each participant's data. Besides, by leveraging differential privacy, we prevent adversaries from exploiting the learned model to infer customers' sensitive information.
- Third, our blockchain-based system prevents malicious model updates by holding all model updates accountable.

Organization. The rest of the chapter is organized as follows. We introduce our design of the system in Section 5.1. Section 5.2 shows the advantages and disadvantages of our designed system. Section 5.3 presents the experimental results showing that our technique is working. Section 5.4 discusses how we prevent information leakage using differential privacy techniques in our designed system. Then, we conclude the chapter and identify future directions in Section 5.5.

Notations. Notations that appear in the rest of the chapter are summarized in Table 5.1.

5.1 System Design

This section introduces a system designed for smart home appliance manufacturers interested in building a machine learning model using data from customers' home appliances to analyze customers' habits and improve their service and products.

TABLE 5.1: Summary of notations

Symbol	Definition
ϵ	differential privacy budget
$\mathcal{B}$	a batch of training examples
N	batch size
μ	mean value of normalized features
σ	standard deviation of normalized features
σ^2	variance of normalized features
$\mathcal{L}_f$	length of feature
$\mathcal{W}_f$	width of feature
$\mathcal{V}_{i,j,k}$	value at a position $\langle i, j \rangle$ for the feature of image k
$\widetilde{\mathcal{V}}_{i,j,k}$	value at a position $\langle i, j \rangle$ for the feature of image k after batch normalization
$\hat{\mathcal{V}}_{i,j,k}$	value at a position $\langle i, j \rangle$ for the feature of image k after our normalized technique
$\mathcal{R}$	the number of updates
f_{bm}	the number of Byzantine customers
$score(i)$	the sum of Euclidean distances of each customer i's update to the closest $\mathcal{R} - f_{bm} - 2$ updates
Δw	model update
rep	reputation value
rep^{Max}	maximal reputation value
$evaluation$	output of the evaluation function
$average$	average reputation value of the participating customers
low	low evaluation result
$high$	high evaluation result
u	the user u selected by the consensus protocol Algorand [155] as one of leading candidates
τ	expected number of sub-users
p	probability of any coin being chosen
M	all users' total amount of coins
GAN	generative adversarial network

5.1.1 System Overview

This section introduces a system designed for smart home appliance manufacturers interested in building a machine learning model using data from customers' home appliances to analyze customers' habits and improve their service and products.

Figure 5.1 illustrates the architecture of our proposed system. The system consists of three primary components: manufacturers, customers, and blockchain. Specifically, manufacturers raise a request for a crowdsourcing FL task. Then, customers who are interested in the crowdsourcing FL tasks submit their trained models to

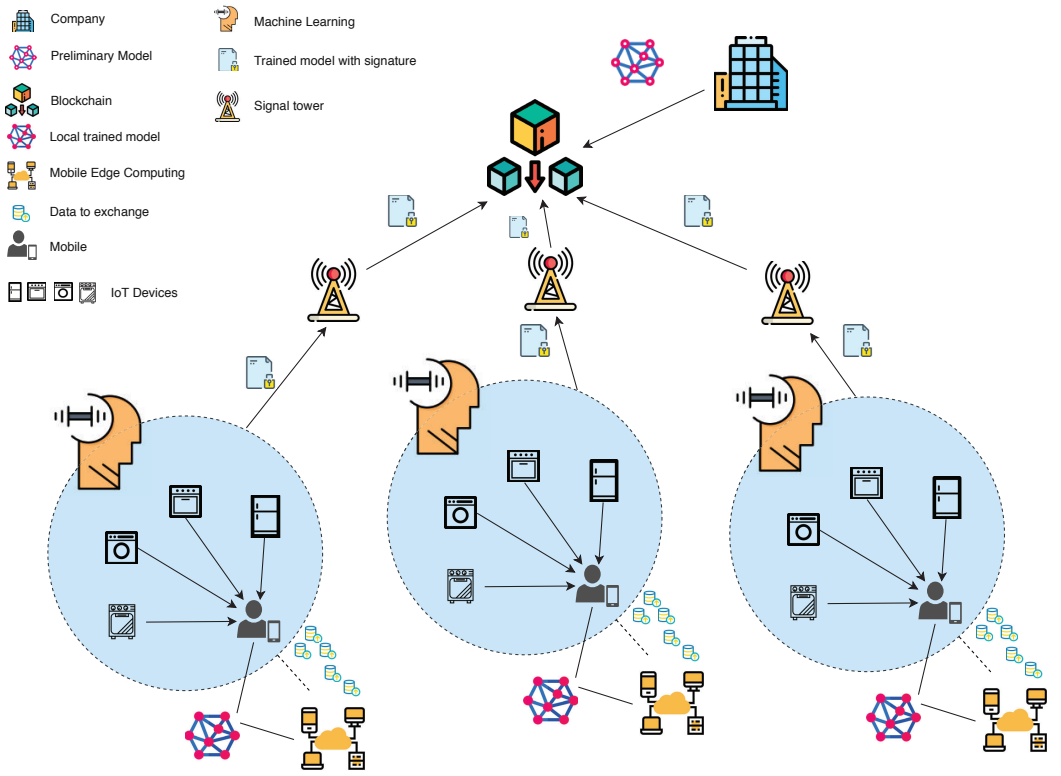


FIGURE 5.1: An overview of our system.

the blockchain. Finally, the blockchain serves as the centralized server to gather customers' models, and a selected miner calculates and generates the global FL model for home appliance manufacturers. Then, we would like to introduce each component in detail.

Manufacturers. Manufacturers raise a request to build a machine learning model to predict customers' consumption behaviours and improve home appliances, which is a crowdsourcing FL task. Customers who have home appliances can participate in the FL task. To facilitate the progress of FL, we use the blockchain to store the initial model with randomly selected parameters. Otherwise, manufacturers need to send the model to everyone or save it in third-party cloud storage. In addition, neither manufacturers nor customers can deny recorded contributions or activities. Eventually, manufacturers will learn a machine learning model as more and more customers participate in the crowdsourcing FL task.

Customers. Customers who have home appliances satisfying crowdsourcing requirements can apply for participating in the FL task. However, since home appliances are equipped with heterogeneous storage and computational powers, it isn't easy to enable each IoT device to train the deep learning model. To address this

issue, we adopt the partitioned deep learning model training approach [85, 156]. Specifically, we use a mobile phone to collect data from home appliances and extract features. To preserve privacy, we add ϵ-DP noise to features. Then, customers continue training fully connected layers in the MEC server. To be specific, we clarify the customers' responsibilities in four detailed steps as follows.

Step 1: Customers download the initial model from the blockchain. Customers who are willing to participate in the FL task check and download the initial model, which is uploaded by the manufactures and available on the blockchain.

Step 2: Customers extract features on the mobile. The mobile phone collects all participating home appliances' data periodically. Then, customers can start training the model using collected data. Since a third party provides the MEC server, it may leak information. Therefore, we divide the local training process into two phases: mobile training and MEC server training. Because perturbing original data directly may compromise the model's accuracy, we treat the convolutional neural network (CNN) layers as the feature extractor to extract features from the original data in the mobile. Then, we add ϵ-DP noise to features before offloading them to the fully connected layers in the MEC.

Step 3: Customers train fully connected layers in the mobile edge computing server. The mobile sends the privacy-preserving features and original labels to the mobile edge computing server so that the server helps train the fully connected layers. The training loss is returned to the mobile to update the front layers.

Step 4: Customers upload models to the blockchain. After training the model, customers sign on hashes of models with their private keys, and then they upload models to the blockchain via smartphones. However, if miners determine that the signature is invalid, the transaction fails because an adversary uses faked data to attack the learning process. After miners confirm the transaction, customers can use the transaction history as an invoice to claim reward and reputation. Section 5.1.2 shows the detail of reputation calculation. By using the immutable property of the blockchain, both manufacturers and customers cannot deny transactions stored on the blockchain.

Blockchain. A consortium blockchain is used in our crowdsourcing system to store machine learning models permanently. The consensus protocol is the Algorand which is based on proof of stake (PoS) as well as byzantine fault tolerance

(BFT) [155, 157]. Algorand relies on BFT algorithms for committing transactions. The following steps are required to reach the consensus: (1) Miners compete for the leader. The ratio of a miner's stake (i.e., coins) to all tokens determines the probability for the miner to be selected. Subsequently, by hashing the output of a random function with the identities of nodes specified by their stake, the order of the block proposals is obtained. Thus, a miner with more stakes will gain a higher chance to become a leader. (2) Committee members verify the block generated by the selected leader. When more than 2/3 of the committee members sign and agree on the leader's block, the new block gets admitted. (3) Committee members execute the gossip protocol to broadcast the new block to neighbours to arrive at a consensus in the blockchain.

In our case, the workflow starts with a manufacturer uploading an initial model to the blockchain. Then, customers can send requests to obtain that model. After training models locally, customers upload their locally trained models to the blockchain. Because of the limitation of the block size, we propose to use IPFS as the off-chain storage. Then, customers upload their models to the IPFS, and a hash will be sent to the blockchain as a transaction. The hash can be used to retrieve the actual data from IPFS. The leader and miners are responsible for confirming transactions and calculating the averaged model parameters to obtain a global model. Miners' results are mainly used for verifying the leader's result. After all, customers upload their trained models, the miners download them and start calculating the averaged model parameters. Then, one of the miners is selected as the leader to upload the global model to the blockchain. We will explain the process in detail as follows:

① *Miners verify the validity of the uploaded model.* When a customer uploads a model or the hash of the model to the blockchain, a miner checks the digital signature of the uploaded file. If the signature is valid, the miner confirms that the update is from the legal participant and puts the transaction in the transaction pool. Subsequently, selected miners constitute a committee to verify all transactions in the pool using Multi-KRUM [158, 159], and accept legitimate updates. After confirming the validity of the uploaded model, the leader selected from miners will generate a new block containing the uploaded file.

② *A selected leader updates the model.* A leader is selected from a group of miners to update the model. Miners compete for updating parameters to get the

reward. Algorand uses the Verifiable Random Functions (VRF) as a local and non-interactive way to select a subset of users as the leading candidates. They form a committee (weighed by their coins) and determine their priorities. A leader candidate with the highest priority will become the leader to update the model parameters. As their coins weigh each user, one coin unit can be regarded as a sub-user. A user with m coins has m "sub-users". Let τ be the expected number of sub-users that the system desires to choose, and M be all users' total amount of coins. Then the probability p of any coin being chosen can be set as τ/M. A user, u with m units of currency, will first use its secret key to generate a hash and proof via VRF. The interval $[0, 1]$ is divided into $m + 1$ sub-intervals, so that the number j of selected sub-users for this user is determined by which sub-interval $hash/2^{hashlen}$ falls in ($hashlen$ denotes the length of $hash$); i.e., j satisfies $hash/2^{hashlen} \in [\sum_{k=0}^{j} \binom{m}{k} p^k (1-p)^{m-k}, \sum_{k=0}^{j+1} \binom{m}{k} p^k (1-p)^{m-k})$ (if $hash/2^{hashlen} = 1$, then $j = m$). Other users can use the proof to check that user u indeed has j sub-users selected. The number of selected sub-users is each user's priority. The user with the highest priority will become the leader. The selected leader is responsible for aggregating models submitted by customers and uploading the global model to the blockchain.

5.1.2 Incentive mechanism

To attract more customers to contribute to building the FL model, we design an incentive mechanism. Because data in home appliances contain confidential information and training consumes computing resources, some customers are unwilling to train the FL model. However, with an incentive mechanism, customers will be rewarded based on their contributions. Then, customers may trade for services, such as the maintenance and upgrade services for appliances, provided by manufacturers using rewards. Specifically, by combining the Multi-KRUM [158, 159] and the reputation-based incentive protocols [160], an incentive mechanism is designed to prevent the poisoning attack as well as reward contributors properly.

After the local model is uploaded, verifiers calculate the reputation using the Multi-KRUM algorithm and eliminate unsatisfied updates. The verifiers, selected based on the VRF [155] from miners, will remove malicious updates by executing the

Multi-KRUM algorithm on updates in the received pool and accept the top majority of the updates received every global epoch. The verifier will add up Euclidean distances of each customer i's update to the closest $\mathcal{R}-f_{bm}-2$ updates and denote the sum as each customer i's score $score(i)$. $\mathcal{R}$ means the number of updates, and f_{bm} implies the number of Byzantine customers. Δw means the model update. It is given by

$$score(i) = \sum_{i \to j} \|\Delta w_i - \Delta w_j\|^2, \tag{5.1}$$

where $i \to j$ denotes the fact that Δw_j belongs to the $\mathcal{R} - f_{bm} - 2$ closest updates to Δw_i. The $\mathcal{R} - f_{bm}$ customers who obtain the lowest scores will be chosen while rejecting the rest.

The value of the reward is proportional to the customer's reputation. If verifiers accept a customer's update, the value of reputation increases by 1; otherwise, it decreases by 1. Each participant is assigned with an initial reputation value *rep*, and *rep* is an integer selected from the $set(0, 1, \cdots, rep^{Max})$, where rep^{Max} denotes the highest reputation. If a miner verifies a solution is correct and provides a positive evaluation, the participant's reputation will increase and be recorded in the blockchain. Let *evaluation* denote the evaluation function's output. $evaluation = high$ denotes a high evaluation result while $evaluation = low$ denotes a low evaluation result. Therefore, the update rule of the reputation *rep* is as follows:

$$rep = \begin{cases} \min(rep^{Max}, rep + 1), & \text{if } evaluation = high \text{ and } rep \geq average, \\ rep - 1, & \text{if } evaluation = low \text{ and } rep \geq average + 1, \\ 0, & \text{if } evaluation = low \text{ and } rep = average, \\ rep + 1, & \text{if } rep < average, \end{cases} \tag{5.2}$$

where *average* denotes the average reputation of the whole customer, and it is the threshold of the selected social strategy, which uses social norms (i.e., Multi-KRUM algorithm) to control customers' behaviours [160]. If a customer's reputation is *average* and receives a *low* feedback after evaluation, her reputation will fall to 0. The blockchain records the status of customers' reputations.

5.1.3 Normalization Technique

To protect the privacy of users' updates, we perturb extracted features in the normalization layer. Now, we present the improvement for the normalization technique proposed in [85]. Although CNN has many channels, our analysis below focuses on one channel only for simplicity. For this channel, suppose the output of the convolutional layers has dimension $\mathcal{L}_f \times \mathcal{W}_f$. Let the value at a position $\langle i, j \rangle$ for the feature of image k be $\mathcal{V}_{i,j,k}$. Given i and j, Jiang *et al.* [85] adopt the batch normalization which transforms $\mathcal{V}_{i,j,k}$ to $\widetilde{\mathcal{V}}_{i,j,k}$, so that for each batch $\mathcal{B}$, the values $\widetilde{\mathcal{V}}_{i,j,k}$ for $k \in \mathcal{B}$ have a mean of 0 and a variance of 1; i.e.,

$$\frac{1}{|\mathcal{B}|} \sum_{k \in \mathcal{B}} \widetilde{\mathcal{V}}_{i,j,k} = 0,$$

while

$$\frac{1}{|\mathcal{B}|} \sum_{k \in \mathcal{B}} (\widetilde{\mathcal{V}}_{i,j,k})^2 = 1.$$

From $|\mathcal{B}| = N$ and the Cauchy–Schwarz inequality, [85] bounds

$$\widetilde{\mathcal{V}}_{i,j,k} \in [-\sqrt{N-1}, \sqrt{N-1}]$$

for any i, j, k, so that if one value in the feature

$$\{\mathcal{V}_{i,j,k} \mid i \in \{1, 2, \ldots, \mathcal{L}_f\} \text{ and } j \in \{1, 2, \ldots, \mathcal{W}_f\}\}$$

of image k varies, the sensitivity of

$$\{\widetilde{\mathcal{V}}_{i,j,k} \mid i \in \{1, 2, \ldots, \mathcal{L}_f\} \text{ and } j \in \{1, 2, \ldots, \mathcal{W}_f\}\}$$

is at most $2\sqrt{N-1}$.

Then, according to `Laplace` mechanism [7], the independent zero-mean Laplace noise with scale $2\sqrt{N-1}/\epsilon$ is added to each $\widetilde{\mathcal{V}}_{i,j,k}$ for $i \in \{1, 2, \ldots, \mathcal{L}_f\}$ and $j \in \{1, 2, \ldots, \mathcal{W}_f\}$ to protect $\mathcal{V}_{i,j,k}$ under ϵ-differential privacy. In our approach, we normalize $\mathcal{V}_{i,j,k}$ for $i \in \{1, 2, \ldots, \mathcal{L}_f\}$ and $j \in \{1, 2, \ldots, \mathcal{W}_f\}$ as

$$\hat{\mathcal{V}}_{i,j,k} \in [-\sqrt{N-1}, \sqrt{N-1}],$$

so that if one value in the feature

$$\{\mathcal{V}_{i,j,k} \mid i \in \{1, 2, \ldots, \mathcal{L}_f\} \text{ and } j \in \{1, 2, \ldots, \mathcal{W}_f\}\}$$

of image k varies, the sensitivity of

$$\{\hat{\mathcal{V}}_{i,j,k} \mid i \in \{1, 2, \ldots, \mathcal{L}_f\} \text{ and } j \in \{1, 2, \ldots, \mathcal{W}_f\}\}$$

is $2\sqrt{N-1}$. Then, based on `Laplace` mechanism [7], the independent zero-mean Laplace noise with scale $2\sqrt{N-1}/\epsilon$ is added to each $\hat{\mathcal{V}}_{i,j,k}$ for $i \in \{1, 2, \ldots, \mathcal{L}_f\}$ and $j \in \{1, 2, \ldots, \mathcal{W}_f\}$ to protect $\mathcal{V}_{i,j,k}$ under ϵ-differential privacy. From the above discussions, batch normalization of [85] enforces not only

$$\widetilde{\mathcal{V}}_{i,j,k} \in [-\sqrt{N-1}, \sqrt{N-1}]$$

but also the mean is

$$\frac{1}{|\mathcal{B}|} \sum_{k \in \mathcal{B}} \widetilde{\mathcal{V}}_{i,j,k} = 0$$

and the variance is

$$\frac{1}{|\mathcal{B}|} \sum_{k \in \mathcal{B}} (\widetilde{\mathcal{V}}_{i,j,k})^2 = 1,$$

while our normalization technique requires only

$$\hat{\mathcal{V}}_{i,j,k} \in [-\sqrt{N-1}, \sqrt{N-1}]$$

without any constraints on the mean and variance. Experiments to be presented in Section 5.3 show that our normalization technique significantly improves the learning accuracy over that of [85].

Next, we explain why our normalization technique outperforms batch normalization. Both Jiang *et al.*'s solution [85] and our solution add the same zero-mean Laplace noise to normalized layer inputs. When using batch normalization, the mean of features $\mu = 0$ and the variance $\sigma^2 = 1$. For ease of explanation, below, we use a Gaussian distribution as an example for the distribution of the features since Gaussian distributions appear in many real-world applications. Note that the actual distribution of the features may not follow Gaussian. According to the three-sigma rule of Gaussian distribution [161], about 99.73% values lie within three standard deviations of the mean. Similarly, most feature values after batch

normalization lie in $[-3\sigma, 3\sigma]$ which is $[-3, 3]$ instead of $[-\sqrt{N-1}, \sqrt{N-1}]$. In contrast, feature values lie more evenly in $[-\sqrt{N-1}, \sqrt{N-1}]$ when using our normalization technique. Thus, features have smaller magnitudes when using batch normalization than using our normalization technique.

In the case of batch normalization, we have

$$\begin{aligned} &\frac{2\sqrt{N-1}}{\epsilon} \gg 3\sigma, \\ \implies &\frac{2\sqrt{N-1}}{\epsilon} \gg 3, \\ \implies &\epsilon \ll \frac{16}{3} \approx 5.33. \end{aligned}$$

Thus, true feature values will be seriously perturbed by noise when the privacy parameter $\epsilon \ll 5.33$ using batch normalization. However, when we use our normalization technique, we obtain that

$$\begin{aligned} &\frac{2\sqrt{N-1}}{\epsilon} \gg \sqrt{N-1}, \\ \implies &\epsilon \ll 2. \end{aligned}$$

Hence, when the privacy parameter $\epsilon \ll 2$, the true value is overwhelmed by the noise. The more significant privacy parameter means less noise, so feature values using batch normalization are more vulnerable. Thus, our above example implies that features will be perturbed more seriously when using batch normalization than our normalization technique. Summarizing above, the trained model with our normalization technique will achieve a higher test accuracy than trained using batch normalization.

5.2 Pros and Cons of our framework

We discuss the advantages and disadvantages of our framework in this section.

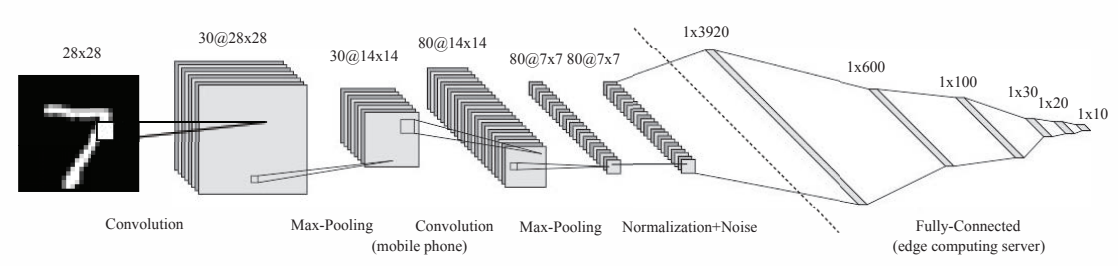


FIGURE 5.2: The neural network used in experiments.

5.2.1 Privacy and Security

Our system leverages differential privacy techniques to protect the privacy of the extracted features. Thus, the system keeps the participating customers' data confidential. Furthermore, the trained model is encrypted and signed by the sender to prevent the attackers and imposters from stealing the model or deriving original data through reverse-engineering.

5.2.2 Delay Crowdsourcing

Assume there are many customers, and the system highly depends on customers' training results to obtain the predictive model in one global epoch. Unlike other crowdsourcing jobs, manufacturers in our system prefer customers to follow their lifestyle instead of rushing to finish the job to obtain the actual status. As a result, customers who seldom use devices may postpone the overall crowdsourcing progress. This problem can be mitigated by using incentive mechanisms. Yu *et al.* [162] designed a queue to store customers who submitted their models in order. Thus, customers who submit their locally trained models early will be rewarded to encourage people to submit their updates earlier.

5.3 Experiments

To validate the effectiveness of our designed FL with a differential privacy approach, we conduct experiments on the MNIST handwritten image dataset [163].

5.3.1 Experiment Setup

The MNIST dataset includes 50,000 training image samples and 10,000 test image samples. Each sample is a 28×28 grayscale image showing a handwritten number within 0 to 9. In addition, the MNIST is a standard dataset employed for testing machine learning algorithms. It gives moderate and typical complexity faced by IoT applications. Therefore, we leverage the MNIST dataset, which has been used for testing the performance of the IoT system by [85, 164–169]. Our designed CNN network includes hidden layers responsible for feature extraction and fully connected layers for classification. Two convolutional layers contain 30 and 80 channels, respectively. A max-pooling layer is deployed to reduce spatial dimensions of the convolutional layers' output after each convolutional layer. Therefore, max-pooling layers accelerate the learning speed of the neural network. Normalization is used after all non-linear layers, i.e., convolutional layers. The normalization layer enables the computation of sensitivity in differential privacy to determine the amount of noise to add, speeds up the learning rate and regularizes gradients from distraction to outliers. Then, we apply ϵ-DP noise to perturb the output of normalization layers to preserve the privacy of the extracted features.

The perturbed features serve as inputs of fully connected layers for classification in the MEC server. In our designed model, fully connected layers include four hidden layers. The dimensions decrease from 3920 to the dimension of the label, which is 10. Finally, there is a softmax layer to predict label and compute loss. The architecture of CNN is shown in Figure 5.2. We simulate FL by constructing the model using the averaged parameters of multiple locally trained model parameters.

In our experiment, we set the hyperparameters of CNN as follows. The learning rate is 0.01, and the batch size N is 64. Then, we set the range of privacy parameter ϵ to be $[1, 10]$. The default number of global epochs is 2, and the default number of local epochs is 40. We use ten participants in the experiment. Before training, we separate the training image dataset into equally ten parts, meaning that each participant gets 6000 training images randomly. We normalize each dimension of the feature to the interval $[-\sqrt{N-1}, \sqrt{N-1}]$ for N denoting the batch size, so that the sensitivity of the normalized feature vector when one dimension of the feature changes is $2\sqrt{N-1}$. Then, according to `Laplace` mechanism [7], the independent zero-mean Laplace noise with scale $2\sqrt{N-1}/\epsilon$ is added to each dimension of the normalized features to protect features under ϵ-differential privacy. Default $\epsilon = 2$.

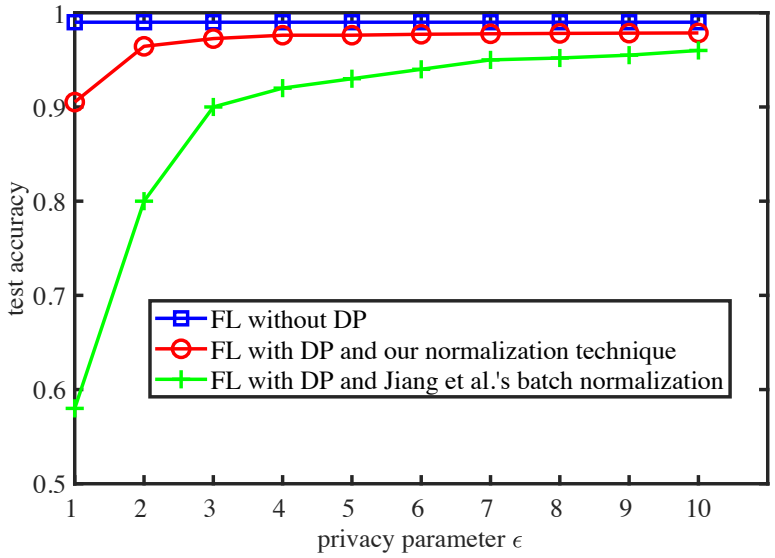


FIGURE 5.3: Impacts of normalization techniques on the test accuracy.

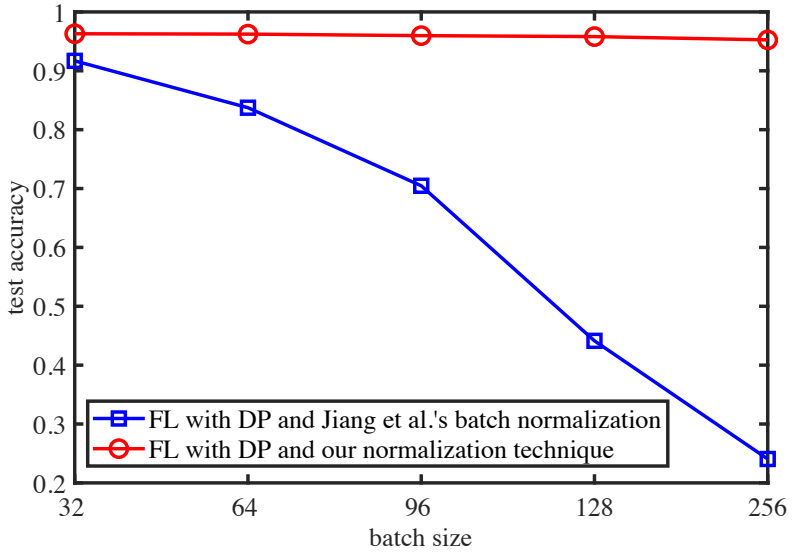


FIGURE 5.4: Impact of the batch size on the test accuracy of the FL model protected with DP ($\epsilon = 2$).

5.3.2 Experimental Results

Figure 5.3 compares the test accuracies between federated learning (FL) without differential privacy (DP) and different DP-aware FL algorithms, including DP-aware FL using our normalization technique and DP-aware FL using Jiang *et al.*'s batch normalization [85]. Figure 5.3 shows the superiority of DP-aware FL using our normalization technique over DP-aware FL using Jiang *et al.*'s batch normalization [85]. Thus, we confirm that our normalization technique is proper when we add the Laplace noise to features because we relax constraints of normalization

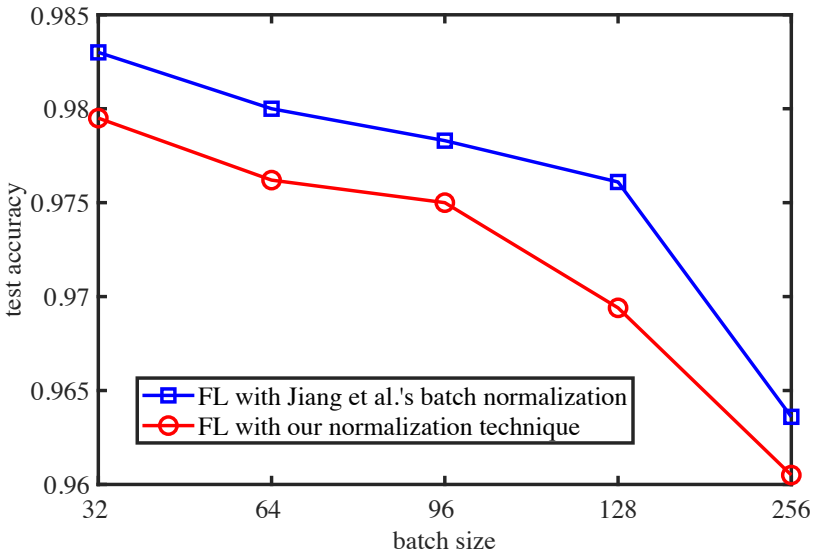


FIGURE 5.5: Impact of the batch size on the test accuracy of the FL model using our normalization technique without DP protection.

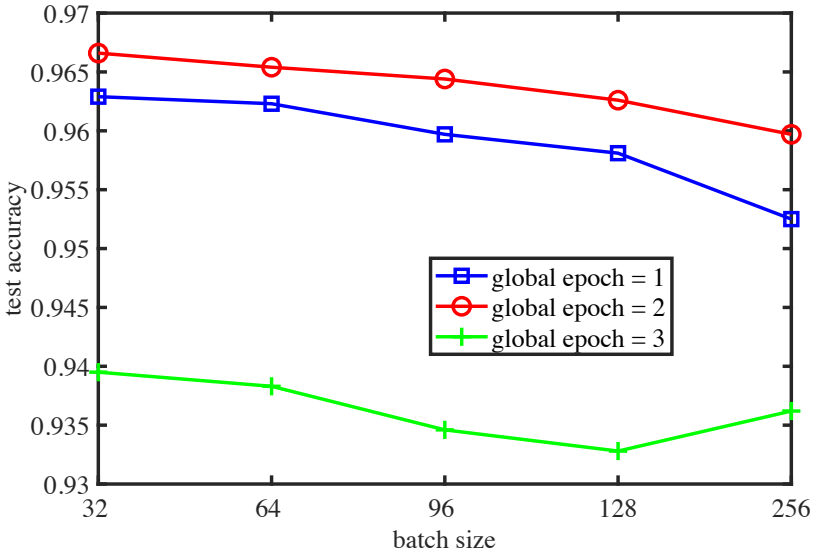


FIGURE 5.6: Impact of the batch size on the test accuracy under different global epochs using our normalization technique ($\epsilon = 2$).

compared with batch normalization as stated in Section 5.1.1. A feature that goes through batch normalization often results in a smaller magnitude than that goes through our normalization technique, so the value of the feature is easily overwhelmed by the noise when using batch normalization. For each DP-aware FL, we also observe that the test accuracy gets closer to the test accuracy of FL without DP as the privacy parameter ϵ increases because a more significant privacy parameter ϵ means less privacy protection which equals that less noise is used. Thus, we conclude that our normalization outperforms batch normalization under

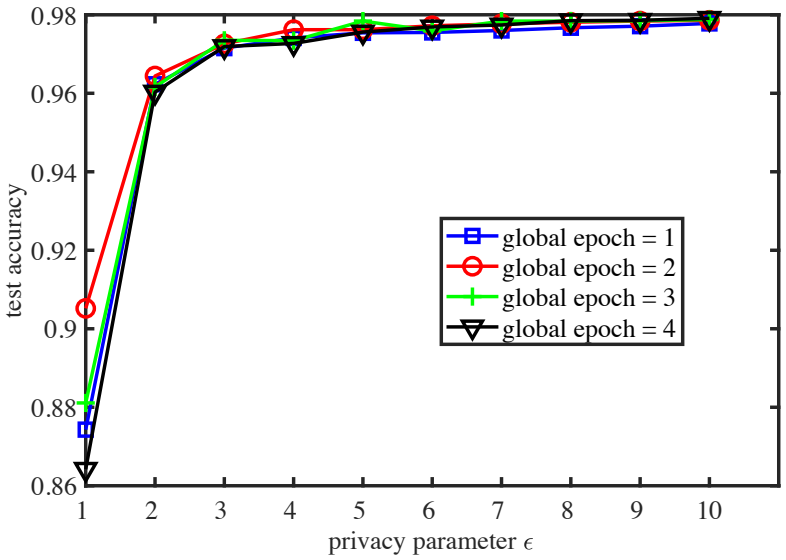


FIGURE 5.7: Impact of DP parameter ϵ on the test accuracy using our normalization technique under various global epochs.

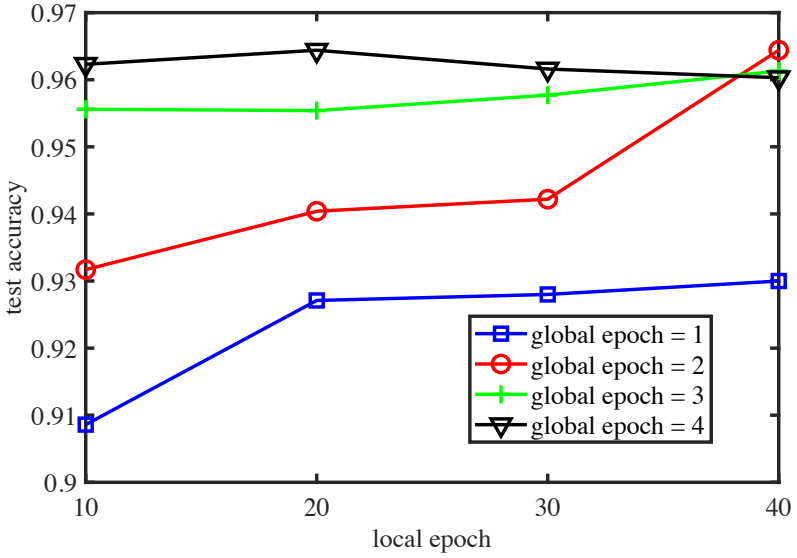


FIGURE 5.8: Impact of the number of local epochs on the test accuracy using our normalization technique under various global epochs when $\epsilon = 2$.

ϵ-differential privacy when training the FL model.

Figure 5.4 presents that the test accuracy of the FL model decreases as the batch size increases when the number of the global epoch is 1 and DP parameter $\epsilon = 2$. This is because we add Laplace noise to features; the added noise will increase as the batch size N increases, resulting in worse test accuracy. Moreover, due to the three-sigma rule in Gaussian distribution, most feature values normalized with batch normalization lie in $[-3\sigma, 3\sigma]$. But feature values normalized using our

normalization technique lie in $[-\sqrt{N-1}, \sqrt{N-1}]$. However, Figure 5.5 shows that if no differential privacy noise is added, the test accuracy with the batch normalization outperforms that using our normalization technique. Moreover, as the batch size increases, the test accuracy will decrease. Therefore, we conclude that our normalization technique works better with FL under DP protection.

Furthermore, Figure 5.6 illustrates that the test accuracy is better when the number of global epochs = 2 than the number of global epoch = 1 or 3 when $\epsilon = 2$ and the number of local epochs is 40. As the number of global epochs increases, the test accuracy increases if DP noise is not added. However, Laplace noise increases as the number of global epochs increases, negatively affecting test accuracy. Thus, a trade-off between the number of global epochs and the amount of noise is required. In our case, when the privacy parameter $\epsilon = 2$ and the number of local epochs is 40, the optimal number of global epochs is 2.

Figure 5.7 illustrates how the privacy parameter ϵ affects the test accuracy of the FL model. In our experiment, we train FL with 4 global epochs to validate the practicality of our designed approach. The test accuracy increases as the privacy parameter ϵ increases. A larger ϵ means that less noise is added to features so that the privacy protection is weaker. Typical ϵ values for experiments are between 0.1 and 10 [153]. Our experiment shows that we can achieve at least 90% accuracy when the global epochs = 2 and the privacy parameter $\epsilon > 1$. Before training, we initialize the model with random parameters, and all parties will use the model with initial parameters for their local training. After the first global epoch, we obtain a new model by averaging all parties' model parameters. Then, in the second global epoch, parties start training using the model from the first global epoch. Through our experiment, we can verify that our designed FL method is effective. However, when the number of global epochs increases to 3 or 4, the test accuracy may decrease. The test accuracy decreases because the noise increases as the number of global epochs increases.

Figure 5.8 shows both the number of local epochs and the number of the global epochs that affect the test accuracy of an FL model. The number of local epochs reflects the cost of devices' computing resources locally. We add ϵ-differential privacy noise during training, and the test accuracy may drop if too much noise is added in each epoch. From Figure 5.8, when the number of local epochs equals 20 or 30, it takes 4 global epochs to achieve similar accuracy. When the number

of local epochs is 40, it takes 2 global epochs. But the test accuracy will start to drop if the number of local epochs is 40 and the number of the global epoch is more than 2. Hence, to obtain a high test accuracy, it necessities optimal values to strike a good balance between the number of local epochs and global epochs for averaging locally uploaded models, which we leave as future work.

5.3.3 Performance evaluation on the mobile device and edge server

Now, we evaluate the feasibility and efficacy of training on mobile devices. A Raspberry Pi 4 Model B tiny computer in Figure 5.9 is used to simulate the mobile device. Key specifications of the Raspberry Pi 4 Model B are listed in Table 5.2. We leverage a laptop to emulate the edge server equipped with four 2.3 GHz Intel Core i5 processors, 8 GB of RAM, and MacOS 10.14.4 system.

TABLE 5.2: Raspberry Pi 4 Model B Specifications [19].

Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @1.5GHz
4GB LPDDR4-3200 SDRAM

In our experiment, we distribute the MNIST dataset [163] with 60,000 images to ten participants equally so that each participant (i.e., each device) has 6,000 images. Then, we run the same training process on both the mobile device and the edge server. It takes about 144 seconds to train the model with 6,000 images on the Raspberry Pi 4 (i.e., the mobile device) for each epoch, and it uses about 9 seconds to train the model on the laptop (i.e., the edge server). For default forty epochs, the mobile device and the edge server use about 96 minutes and 6 minutes, respectively. A client is supposed to participate in the federated learning when the smartphone is idle, such as charging, screen off, and connected to an unmetered network, for example, WiFi [32, 170]. Thus, we confirm that it is feasible to utilize mobile devices in the federated learning. Besides, an edge server will significantly improve the speed of training because it trains much faster.

In addition to the training time, the delay of our proposed approach, which depends on the transmission rate, is small because smartphones often use wideband network connections (e.g., 4G and WiFi). The average size of locally trained models is 617.8KB in our experiment. Assume the upload bandwidth is 1MB/s, so the

FIGURE 5.9: Raspberry Pi 4 Model B.

communication cost is 0.6178 second. The communication cost is little compared with wasted training time on the mobile device.

5.3.4 Evaluation on the incentive mechanism

In this section, we evaluate the impacts of the incentive mechanism on customers' rewards and reputation. The assumption and parameters in the experiments are as follows. Assume that the maximum values of both reputation and reward are 100 (i.e., $rep^{Max} = 100$). Every customer has a reputation of 5 (i.e., $average = 5$) at the beginning. We set the reward for each accepted update equal to the owners' reputation in each global epoch. The experiments compare reward and reputation that customers can achieve in four cases (i.e., no incentive mechanism, honest customer, the malicious customer performs poisoning attack at global epoch = 1, and malicious customer performs poisoning attack at global epoch = 4). If there is no incentive mechanism, the customer gets a fixed reward of 5 in every global epoch.

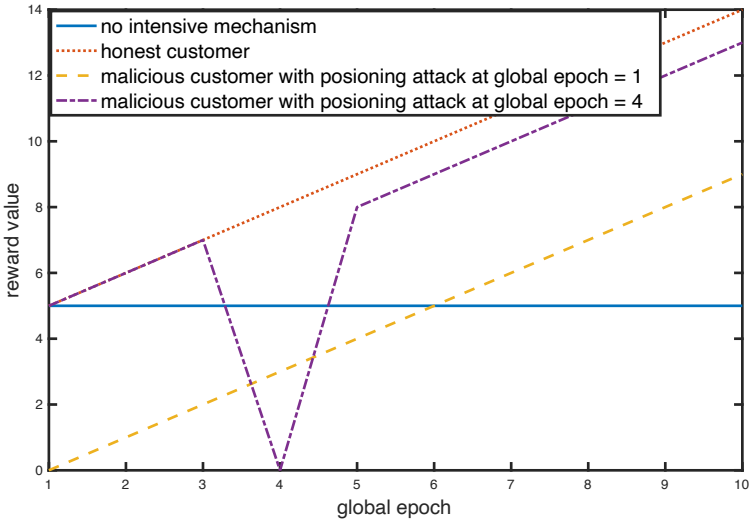


FIGURE 5.10: Reward comparison.

As shown in Figure 5.10, when there is no incentive mechanism, the reward value is the same in each global epoch regardless of poisoning updates. However, with the incentive mechanism, the honest customer, whose updates are accepted, will gain more rewards as the number of global epochs increases. If a customer's update is considered as poisoning (i.e., the value of s in Eq. (5.1) is significantly larger than others), her update will not be accepted; that is, her reward is 0. Besides, the behaviour of the poisoning attack affects the value of reputation, which results in a decrease in reputation. If the poisoning attack is performed when the value of the reputation is equal to the *average*, the customer's reputation will be clear, which will result in small rewards afterward. However, if the malicious behaviour happens when the reputation value is higher than *average*, the reputation drops by 1, so does the reward in the subsequent global epoch.

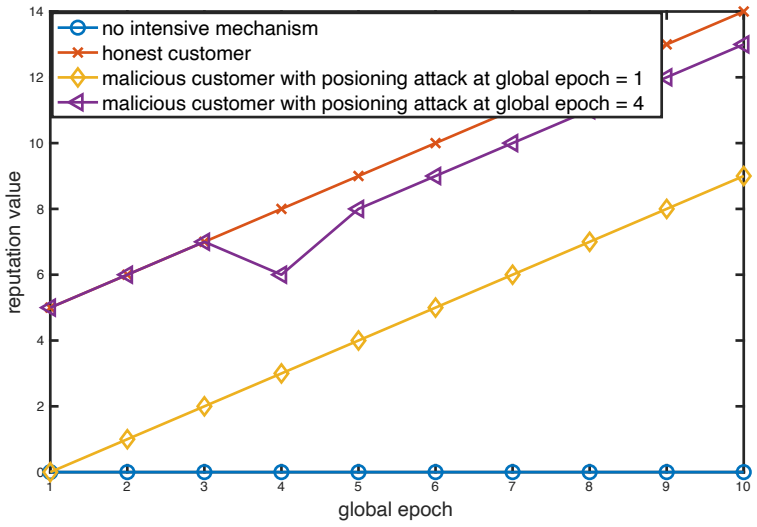


FIGURE 5.11: Reputation comparison.

Figure 5.11 shows the impact of the incentive mechanism on reputation. Without the incentive mechanism, customers' reputations will be 0. If a customer is honest and uploads the correct update in every global epoch, her reputation increases as the global epoch increases. However, if a customer uploads a malicious update when her reputation value equals the *average* (i.e., 5), her reputation will drop to 0. However, if her reputation is not 5, her reputation drops by 1 when caught performing a poisoning attack.

Thus, our incentive mechanism can encourage honest customers to contribute their valuable updates while preventing malicious customers from performing the poisoning attack.

5.4 Discussion

In order to attract more customers to train the global model, our designed system should guarantee that customers' confidential information will not leak. Studies are discussing potential risks of information leakage in FL [38, 171] in which attackers may infer customers' private data from gradients. We leverage differential privacy to perturb features before classification in the fully connected layers to prevent this scenario. Thus, gradients are also protected by differential privacy. Hitaj *et al.* [38] demonstrated that a curious server could obtain confidential information using the generative adversarial network if gradients were protected by a large privacy budget in the collaborative learning. But their experiments confirmed that GAN-based approaches might not work well when the selected privacy parameter was smaller than 10, which is the upper bound of privacy parameter in our experiment, and we can achieve the accuracy of 97%. Therefore, our designed approach guarantees accuracy and protects the privacy of local models as well as data. In addition, Yin *et al.* [171] introduced a DeepInversion method that could invert a trained neural network to a synthesized class-conditional input image starting from the random noise. However, they leveraged the information stored in the batch normalization layer. In our trained model, we add Laplacian noise during training in the batch normalization layer, and then attackers cannot obtain the true information stored there. Thus, their approach is ineffective against our trained model.

5.5 Conclusion

This chapter presents a blockchain-based crowdsourcing FL system for IoT device manufacturers to learn customers better. We use multiple state-of-the-art technologies to construct the system, including the mobile edge computing server,

blockchain, distributed storage, and federated learning. Besides, our system enforces differential privacy to protect the privacy of customers' data. To improve the accuracy of the FL model, we remove constraints of the batch normalization technique if the privacy of features is protected by differential privacy. By designing a proper incentive mechanism for the crowdsourcing task, customers are more likely to participate in the crowdsourcing tasks. The blockchain will audit all customers' updates during the federated learning so that the system can hold the model updates accountable to prevent malicious customers or manufacturers.

Chapter 6

Conclusion and Future Work

The main topic of this thesis is to study privacy-preserving data analytics technologies. For example, LDP and FL technologies in Chapter 3, and DP and blockchain techniques in Chapter 4. In particular, we focus on ensuring the privacy of data generated from IoT smart devices, such as vehicle location information. This chapter summarizes the main ideas of our studies and points out possible future research directions.

6.1 Conclusion

The development of sensors and communication technologies for IoT has enabled a fast and large-scale collection of user data, and bred new applications; for example, the Waze application provides the intelligent transportation routing service. This kind of service benefits users' daily lives, but it may raise privacy concerns regarding sensitive data such as users' location information. In addition to LDP, centralized DP mechanisms have been widely explored in the literature to protect privacy. The privacy parameter bounds the information about the dataset leaked by the noisy output. Often, a dataset needs to be used for answering multiple queries, so the level of privacy protection may degrade as more queries are answered. Thus, it is crucial to keep track of privacy budget spending, which should not exceed the given limit of the privacy budget. Moreover, if a query has been answered before and is asked again on the same dataset, we may reuse the previous noisy response for the current query to save the privacy cost.

In Chapter 1, we have presented the background of IoT and privacy technologies as well as existing research challenges. Then, we specify the motivations and methodologies of this thesis.

In Chapter 2, we have reviewed the related literature. Moreover, we have highlighted the novel contributions of our studies compared with other existing solutions.

In Chapter 3, we have proposed the `LDP-FedSGD` algorithm for conducting federating learning in IoV as well as four LDP mechanisms. The `LDP-FedSGD` algorithm enables the vehicular crowdsourcing applications to train a machine learning model to predict the traffic status while avoiding the privacy threat and reducing the communication cost. Moreover, we propose `PM-OPT`, `PM-SUB`, `Three-Outputs`, and `HM-TP` mechanisms. These mechanisms effectively preserve privacy when collecting data records and computing accurate statistics in various data analysis tasks, including estimating the mean values on numerical attributes and preserving gradients' privacy in various machine learning tasks. Specifically, by leveraging LDP mechanisms, adversaries are unable to deduce the original data from uploaded gradients. Next, FL enables edge computing nodes to train the local machine learning models by using collected data and sending noisy gradients instead of data to the cloud server to obtain a global model. Furthermore, extensive experiments demonstrate that our proposed approaches are effective and perform better than existing solutions.

In Chapter 4, a blockchain-based approach for tracking and saving differential privacy costs has been proposed. Also, we have designed an algorithm that reuses noise fully or partially for different instances of the same query type to minimize the accumulated privacy cost. The efficiency of the algorithm is proved via rigorous mathematical proof. Moreover, to confirm our proposed approach's effectiveness, we design a blockchain-based system for doing experiments by using real-world datasets.

In Chapter 5, a privacy-preserving system for appliance manufacturers has been proposed to utilize customers' data for training machine learning models. Additionally, we add differential privacy noises after the normalization layer to protect the privacy of the model. To mitigate the effect of differential privacy noises on

the model's accuracy, we propose to remove the mean and variance constraints of the batch normalization while training the deep learning models.

To sum up, in our thesis, we have addressed challenges in privacy-preserving data analytics using DP, LDP, FL, and blockchain technologies. The theories and algorithms we propose are not limited to IoT data. Also, in other areas, as long as privacy-preserving data analytics challenges are addressed, they may refer to our proposed algorithms.

6.2 Future Work

In the following, many open directions are introduced for future work.

6.2.1 Local Differential Privacy for Federated Deep Learning

We apply LDP mechanisms to federated machine learning algorithms for protecting the privacy of data. In the future, we would like to use LDP mechanisms in more sophisticated data analysis tasks, for example, federated deep learning. Like federated machine learning, the user's data privacy leakage problem also exists in deep learning. So, LDP mechanisms can be applied to federated deep learning as well.

The dimension of gradient generated in deep learning is high. Currently, we sample k dimensions from the gradient and submit them to the central aggregator to reduce the effect of high dimension. However, LDP requires millions of users to ensure accuracy. When the dimension is very high, we cannot expect enough population to report to the same dimension, reducing statistical accuracy.

Therefore, how to reduce the data dimension effectively to increase statistical accuracy is quite challenging. We would like to explore more possible dimension deduction methods in federated deep learning and combine them with the LDP mechanism to train a more accurate model while maintaining an LDP guarantee.

6.2.2 (ϵ, δ)-Local Differential Privacy Mechanisms

In Chapter 3, we propose `Three-Outputs`, `PM-OPT`, and `PM-SUB` mechanisms satisfying ϵ-LDP. A natural extension is to extend proposed mechanisms to satisfy (ϵ, δ)-LDP. (ϵ, δ)-DP is proposed by [172] which is a relaxation variant of ϵ-DP [6]. (ϵ, δ)-LDP achieves the better accuracy than ϵ-LDP when applied to machine learning algorithms. Gaussian mechanism and optimal Gaussian mechanism [173] which satisfy (ϵ, δ)-DP can be applied to (ϵ, δ)-LDP directly.

6.2.3 Novel Normalization Technique for Privacy-Preserving Deep Learning

In Chapter 5, we add differential privacy noises after the normalization layer to protect the privacy of extracted features. We remove the constraints of mean and variance in batch normalization to enhance the model accuracy. However, our approach is unable to solve the vanishing gradient problem fundamentally. In addition, noises increase dramatically with the increase of the number of global epochs after two epochs. The optimal trade-off between global epochs and differential privacy noises worthies future work to determine. In order to preserve privacy and improve the model accuracy, a novel normalization technique should be proposed to resolve above two challenges.

6.2.4 Novel Local Differential Privacy Mechanisms

The approach of designing `Four-Outputs` is similar to `Three-Outputs`, but the detailed analysis for `Four-Outputs` will be even more tedious than that for `Three-Outputs` (which is already quite complicated). Given above reasons, we elaborate `Three-Outputs` but not `Four-Outputs` in Chapter 3. However, as the privacy budget increases, it is encouraged to send as much information as possible. Thus, there will be an LDP mechanism with four output possibilities to ensure the best performance (i.e., smallest worst-case variance) at some range of privacy parameters. Similarly, the LDP mechanism, which has five outputs or six output possibilities, will get the smallest worst-case variance with some privacy parameters. Therefore, my future work is to develop new LDP mechanisms with four

output or five output, etc., possibilities to obtain a smaller worst-case variance than that of existing mechanisms.

The challenge in developing a new mechanism is the complicated mathematical analysis. Thus, it is challenging to obtain the worst-case variance due to the comparison and discussion of boundaries for the worst-case variance expression as the privacy parameter increases. Besides, finding the parameter to minimize the worst-case variance is complicated because we need to solve a quartic equation and determine the optimal parameters to minimize the worst-case variance in different scopes. When there is a mechanism with four output or five output possibilities, the design approach with four output or five output possibilities is similar to that of the three outputs mechanism. Still, the detailed analysis for the mechanism with four output or five output possibilities will be even more tedious than that for the mechanism with three outputs (which is already quite complicated).

Appendix A

Appendix for Chapter 3

A.1 Proof of Lemma 1

The mechanism $\mathcal{M}_2$ satisfies the proper distribution in requirement (3.8c) because

$$\begin{aligned}
&P_{C\leftarrow x}(\mathcal{M}_2) + P_{-C\leftarrow x}(\mathcal{M}_2) + P_{0\leftarrow x}(\mathcal{M}_2) \\
&= \frac{P_{C\leftarrow x}(\mathcal{M}_1) + P_{-C\leftarrow -x}(\mathcal{M}_1)}{2} \\
&+ \frac{P_{-C\leftarrow x}(\mathcal{M}_1) + P_{C\leftarrow -x}(\mathcal{M}_1)}{2} \\
&+ \frac{P_{0\leftarrow x}(\mathcal{M}_1) + P_{0\leftarrow -x}(\mathcal{M}_1)}{2} = 1.
\end{aligned}$$

Besides, the mechanism $\mathcal{M}_2$ satisfies the unbiased estimation in Eq. (3.8b) because

$$\begin{aligned}
&C \cdot P_{C\leftarrow x}(\mathcal{M}_2) + (-C) \cdot P_{-C\leftarrow x}(\mathcal{M}_2) + 0 \cdot P_{0\leftarrow x}(\mathcal{M}_2) \\
&= C \cdot \frac{P_{C\leftarrow x}(\mathcal{M}_1) + P_{-C\leftarrow -x}(\mathcal{M}_1)}{2} \\
&+ (-C) \cdot \frac{P_{-C\leftarrow x}(\mathcal{M}_1) + P_{C\leftarrow -x}(\mathcal{M}_1)}{2} \\
&+ 0 \cdot \frac{P_{0\leftarrow x}(\mathcal{M}_1) + P_{0\leftarrow -x}(\mathcal{M}_1)}{2} = x.
\end{aligned}$$

In addition, we have

$$\frac{P_{C\leftarrow x}(\mathcal{M}_2)}{P_{C\leftarrow x'}(\mathcal{M}_2)} = \frac{P_{C\leftarrow x}(\mathcal{M}_1) + P_{-C\leftarrow -x}(\mathcal{M}_1)}{P_{C\leftarrow x'}(\mathcal{M}_1) + P_{-C\leftarrow -x'}(\mathcal{M}_1)}, \tag{A.1}$$

$$\frac{P_{-C\leftarrow x}(\mathcal{M}_2)}{P_{-C\leftarrow x'}(\mathcal{M}_2)} = \frac{P_{-C\leftarrow x}(\mathcal{M}_1) + P_{C\leftarrow -x}(\mathcal{M}_1)}{P_{-C\leftarrow x'}(\mathcal{M}_1) + P_{C\leftarrow -x'}(\mathcal{M}_1)}, \tag{A.2}$$

and

$$\frac{P_{0\leftarrow x}(\mathcal{M}_2)}{P_{0\leftarrow x'}(\mathcal{M}_2)} = \frac{P_{0\leftarrow x}(\mathcal{M}_1) + P_{0\leftarrow -x}(\mathcal{M}_1)}{P_{0\leftarrow x'}(\mathcal{M}_1) + P_{0\leftarrow -x'}(\mathcal{M}_1)}. \tag{A.3}$$

According to (3.8a), we obtain

$$\begin{aligned}
&\frac{e^{-\epsilon}(P_{C\leftarrow x'}(\mathcal{M}_1) + P_{-C\leftarrow -x'}(\mathcal{M}_1))}{P_{C\leftarrow x'}(\mathcal{M}_1) + P_{-C\leftarrow -x'}(\mathcal{M}_1)} \\
&\leq Eq.\ (A.1) \leq \frac{e^{\epsilon}(P_{C\leftarrow x'}(\mathcal{M}_1) + P_{-C\leftarrow -x'}(\mathcal{M}_1))}{P_{C\leftarrow x'}(\mathcal{M}_1) + P_{-C\leftarrow -x'}(\mathcal{M}_1)},
\end{aligned}$$

which is equivalent to

$$e^{-\epsilon} \leq \text{Eq. } (A.1) \leq e^{\epsilon}. \tag{A.4}$$

Similarly, we prove that Eq. (A.2) and Eq. (A.3) satisfy (3.8a). Hence, we conclude that $\mathcal{M}_2$ satisfies ϵ-LDP's requirements.

Then, we prove that the symmetrization process does not increase the worst-case noise variance as follows:

Since $\mathcal{M}_2$ satisfies the unbiased estimation, $\mathbb{E}[Y|X = x] = x$. The variance of mechanism $\mathcal{M}_2$ given x is

$$\begin{aligned}
&\text{Var}_{\mathcal{M}_2}[Y|X = x] = \mathbb{E}[Y^2|X = x] - (\mathbb{E}[Y|X = x])^2 \\
&= C^2 \cdot P_{C\leftarrow x}(\mathcal{M}_2) + 0 \cdot P_{0\leftarrow x}(\mathcal{M}_2) \\
&\quad + (-C)^2 \cdot P_{-C\leftarrow x}(\mathcal{M}_2) - x^2 \\
&= C^2(1 - P_{0\leftarrow x}(\mathcal{M}_2)) - x^2 \qquad (A.5) \\
&= C^2\left(1 - \frac{P_{0\leftarrow x}(\mathcal{M}_1) + P_{0\leftarrow -x}(\mathcal{M}_1)}{2}\right) - x^2, \qquad (A.6)
\end{aligned}$$

or it changes to

$$\begin{aligned}
&\mathrm{Var}_{\mathcal{M}_2}[Y|X=-x] = \mathbb{E}[Y^2|X=-x] - (\mathbb{E}[Y|X=-x])^2 \\
&= C^2 \cdot P_{C\leftarrow -x}(\mathcal{M}_2) + 0 \cdot P_{0\leftarrow -x}(\mathcal{M}_2) \\
&\quad + (-C)^2 \cdot P_{-C\leftarrow -x}(\mathcal{M}_2) - x^2 \\
&= C^2(1 - P_{0\leftarrow -x}(\mathcal{M}_2)) - x^2 \qquad \text{(A.7)} \\
&= C^2 \left(1 - \frac{P_{0\leftarrow x}(\mathcal{M}_1) + P_{0\leftarrow -x}(\mathcal{M}_1)}{2}\right) - x^2, \qquad \text{(A.8)}
\end{aligned}$$

when given $-x$.

The variance of mechanism $\mathcal{M}_1$ given x is

$$\begin{aligned}
&\mathrm{Var}_{\mathcal{M}_1}[Y|X=x] = \mathbb{E}[Y^2|X=x] - (\mathbb{E}[Y|X=x])^2 \\
&= C^2 \cdot P_{C\leftarrow x}(\mathcal{M}_1) + 0 \cdot P_{0\leftarrow x}(\mathcal{M}_1) \\
&\quad + (-C)^2 \cdot P_{-C\leftarrow x}(\mathcal{M}_1) - x^2 \\
&= C^2(1 - P_{0\leftarrow x}(\mathcal{M}_1)) - x^2, \qquad \text{(A.9)}
\end{aligned}$$

or

$$\begin{aligned}
&\mathrm{Var}_{\mathcal{M}_1}[Y|X=-x] = \mathbb{E}[Y^2|X=-x] - (\mathbb{E}[Y|X=-x])^2 \\
&= C^2 \cdot P_{C\leftarrow -x}(\mathcal{M}_1) + 0 \cdot P_{0\leftarrow -x}(\mathcal{M}_1) \\
&\quad + (-C)^2 \cdot P_{-C\leftarrow -x}(\mathcal{M}_1) - x^2 \\
&= C^2(1 - P_{0\leftarrow -x}(\mathcal{M}_1)) - x^2, \qquad \text{(A.10)}
\end{aligned}$$

when given $-x$.

Then, we obtain that the variance of mechanism $\mathcal{M}_2$ is smaller than or equal to the variance of mechanism $\mathcal{M}_1$ as follows:

$$\begin{aligned}
&\mathrm{Var}_{\mathcal{M}_2}[Y|X=x] = \mathrm{Var}_{\mathcal{M}_2}[Y|X=-x] \\
&= \frac{\mathrm{Var}_{\mathcal{M}_2}[Y|X=x] + \mathrm{Var}_{\mathcal{M}_2}[Y|X=-x]}{2} \\
&= \frac{\text{Eq. (A.9)} + \text{Eq. (A.10)}}{2} \\
&\leq \max\{\mathrm{Var}_{\mathcal{M}_1}[Y|X=-x], \mathrm{Var}_{\mathcal{M}_1}[Y|X=x]\}.
\end{aligned} \qquad \text{(A.11)}$$

Hence, we conclude that the worst-case variance of mechanism $\mathcal{M}_2$ does not increase. ■

A.2 Proof of Lemma 2

In essence, with Eq. (3.20), we have

$$\begin{aligned}
&\frac{P_{C\leftarrow 1}(\mathcal{M}_3)}{P_{C\leftarrow -1}(\mathcal{M}_3)} \\
&= \frac{P_{C\leftarrow 1}(\mathcal{M}_2) - \frac{e^{\epsilon}P_{C\leftarrow -1}(\mathcal{M}_2) - P_{C\leftarrow 1}(\mathcal{M}_2)}{e^{\epsilon}-1}}{P_{C\leftarrow -1}(\mathcal{M}_2) - \frac{e^{\epsilon}P_{C\leftarrow -1}(\mathcal{M}_2) - P_{C\leftarrow 1}(\mathcal{M}_2)}{e^{\epsilon}-1}} \\
&= e^{\epsilon}.
\end{aligned}$$

Similarly, we can prove that $\frac{P_{C\leftarrow 1}(\mathcal{M}_3)}{P_{C\leftarrow -1}(\mathcal{M}_3)} = \frac{P_{-C\leftarrow -1}(\mathcal{M}_3)}{P_{-C\leftarrow 1}(\mathcal{M}_3)} = e^{\epsilon}$. Besides, mechanism $\mathcal{M}_3$ follows the proper distribution as follows:

$$P_{C\leftarrow x}(\mathcal{M}_3) + P_{-C\leftarrow x}(\mathcal{M}_3) + P_{0\leftarrow x}(\mathcal{M}_3) = 1.$$

In addition, we prove that mechanism $\mathcal{M}_3$ satisfies the unbiased estimation as follows:

$$\begin{aligned}
&C \cdot P_{C\leftarrow x}(\mathcal{M}_3) + (-C) \cdot P_{-C\leftarrow x}(\mathcal{M}_3) + 0 \cdot P_{0\leftarrow x}(\mathcal{M}_3) \\
&= C \cdot \left(P_{C\leftarrow x}(\mathcal{M}_2) - P_{-C\leftarrow x}(\mathcal{M}_2)\right) \\
&= x.
\end{aligned}$$

Hence, mechanism $\mathcal{M}_3$ satisfies requirements in (3.8a) (3.8b) (3.8c).

Because the symmetric mechanism $\mathcal{M}_3$ satisfies requirements in (3.8a) (3.8b) (3.8c), the variance of $\mathcal{M}_3$ is

$$\begin{aligned}
&\mathrm{Var}_{\mathcal{M}_3}[Y|X=x] = \mathbb{E}[Y^2|X=x] - (\mathbb{E}[Y|X=x])^2 \\
&= C^2 \cdot P_{C\leftarrow x}(\mathcal{M}_3) + 0 \cdot P_{C\leftarrow x}(\mathcal{M}_3) + (-C)^2 \cdot P_{-C\leftarrow x}(\mathcal{M}_3) - x^2 \\
&= C^2(P_{C\leftarrow x}(\mathcal{M}_3) + P_{-C\leftarrow x}(\mathcal{M}_3)) - x^2 \\
&= C^2(1 - P_{0\leftarrow x}(\mathcal{M}_3)) - x^2.
\end{aligned}$$

By comparing with the variance of $\mathcal{M}_2$ in Eq. (A.5), we obtain

$$\begin{aligned}&\mathrm{Var}_{\mathcal{M}_3}[Y|X=x] - \mathrm{Var}_{\mathcal{M}_2}[Y|X=x]\\ &= C^2(1 - P_{0\leftarrow x}(\mathcal{M}_3)) - x^2 - (C^2(1 - P_{0\leftarrow x}(\mathcal{M}_2)) - x^2)\\ &= C^2(P_{0\leftarrow x}(\mathcal{M}_2) - P_{0\leftarrow x}(\mathcal{M}_3)).\end{aligned}$$

From Inequality (3.19) and Eq. (3.22), we obtain $P_{0\leftarrow x}(\mathcal{M}_2) < P_{0\leftarrow x}(\mathcal{M}_3)$, so that

$$\mathrm{Var}_{\mathcal{M}_2}[Y|X=x] > \mathrm{Var}_{\mathcal{M}_3}[Y|X=x].$$

Thus, when $x \in [-1, 1]$, the variance of $\mathcal{M}_3$ is smaller than the variance of $\mathcal{M}_2$. So, we obtain that the worst-case noise variance of $\mathcal{M}_3$ is smaller that of $\mathcal{M}_2$ when $x \in [-1, 1]$, i.e.,

$$\max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{M}_2}[Y|X=x] > \max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{M}_3}[Y|X=x].$$

■

A.3 Proof of Lemma 3

As $P_{-C\leftarrow 0} + P_{C\leftarrow 0} + P_{0\leftarrow 0} = 1$ and $-C \cdot P_{-C\leftarrow 0} + C \cdot P_{C\leftarrow 0} + 0 \cdot P_{0\leftarrow 0} = 0$, we have

$$P_{C\leftarrow 0} = P_{-C\leftarrow 0} = \frac{1 - P_{0\leftarrow 0}}{2}. \tag{A.12}$$

Then, based on requirements in (3.8a) (3.8b) (3.8c) and Lemma 2, we can derive C with the following steps:

$$\begin{aligned}P_{-C\leftarrow 1} + P_{C\leftarrow 1} + P_{0\leftarrow 1} &= 1,\\ \text{and } -C \cdot P_{-C\leftarrow 1} + C \cdot P_{C\leftarrow 1} + 0 \cdot P_{0\leftarrow 1} &= 1.\end{aligned}$$

Therefore, we have

$$\begin{aligned}P_{C\leftarrow 1} &= \frac{1 - P_{0\leftarrow 1} + \frac{1}{C}}{2},\\ \text{and } P_{-C\leftarrow 1} &= \frac{1 - P_{0\leftarrow 1} - \frac{1}{C}}{2}.\end{aligned}$$

From Lemma 2, we obtain

$$\frac{1 - P_{0\leftarrow 1} + \frac{1}{C}}{2} = e^{\epsilon} \cdot \left(\frac{1 - P_{0\leftarrow 1} - \frac{1}{C}}{2} \right),$$

which is equivalent to

$$C = \frac{e^{\epsilon} + 1}{(e^{\epsilon} - 1)(1 - P_{0\leftarrow 1})}. \tag{A.13}$$

Hence, we have

$$\begin{aligned} P_{C\leftarrow 1} = P_{-C\leftarrow -1} &= \frac{(1 - P_{0\leftarrow 1})e^{\epsilon}}{e^{\epsilon} + 1}, \\ \text{and } P_{-C\leftarrow 1} = P_{C\leftarrow -1} &= \frac{(1 - P_{0\leftarrow 1})}{e^{\epsilon} + 1}. \end{aligned} \tag{A.14}$$

Then, we compute the variance as follows:

I. For $x \in [0, 1]$, we have

$$\begin{aligned} &\mathrm{Var}[Y|X = x] \\ &= \mathbb{E}[Y^2|X = x] - (\mathbb{E}[Y|X = x])^2 \\ &= C^2 \cdot P_{C\leftarrow x} + 0 \cdot P_{0\leftarrow x} + (-C)^2 \cdot P_{-C\leftarrow x} - x^2 \\ &= C^2 \left(P_{C\leftarrow x} + P_{-C\leftarrow x} \right) - x^2. \end{aligned} \tag{A.15}$$

Substituting Eq. (3.12) and Eq. (3.13) into Eq. (A.15) yields

$$\begin{aligned} &C^2 \left(P_{C\leftarrow 0} + (P_{C\leftarrow 1} - P_{C\leftarrow 0})x \right) \\ &\quad + C^2 \left(P_{-C\leftarrow 0} - (P_{-C\leftarrow 0} - P_{-C\leftarrow 1})x \right) - x^2 \\ &= C^2 \left(P_{C\leftarrow 0} + P_{-C\leftarrow 0} \right) + C^2 (P_{C\leftarrow 1} + P_{C\leftarrow -1})x \\ &\quad - C^2 (P_{C\leftarrow 0} + P_{-C\leftarrow 0})x - x^2 \\ &= C^2 \left(P_{C\leftarrow 0} + P_{-C\leftarrow 0} \right) + C^2 (1 - P_{0\leftarrow 1})x \\ &\quad - C^2 (1 - P_{0\leftarrow 0})x - x^2 \\ &= C^2 \left(1 - P_{0\leftarrow 0} \right) + C^2 (P_{0\leftarrow 0} - P_{0\leftarrow 1})x - x^2. \end{aligned} \tag{A.16}$$

II. For $x \in [-1, 0]$, we have

$$\begin{aligned}
&\mathrm{Var}[Y|X=x] \\
&= \mathbb{E}[Y^2|X=x] - (\mathbb{E}[Y|X=x])^2 \\
&= C^2\left(1 - P_{0\leftarrow 0}\right) + C^2(P_{0\leftarrow 0} - P_{0\leftarrow 1})(-x) - x^2. \qquad \text{(A.17)}
\end{aligned}$$

Hence, by summarizing Eq. (A.13), Eq. (A.16), and Eq. (A.17), we get the variance as follows:

$$\begin{aligned}
&\mathrm{Var}[Y|X=x] \\
&= C^2\left(1 - P_{0\leftarrow 0}\right) + C^2(P_{0\leftarrow 0} - P_{0\leftarrow 1})|x| - x^2 \\
&= \left(\frac{e^\epsilon + 1}{(e^\epsilon - 1)(1 - P_{0\leftarrow 1})}\right)^2 (1 - P_{0\leftarrow 0} + (P_{0\leftarrow 0} - P_{0\leftarrow 1})|x|) - x^2. \qquad \text{(A.18)}
\end{aligned}$$

Derive the partial derivative of $\mathrm{Var}[Y|X=x]$ to $P_{0\leftarrow 1}$, and we get

$$\frac{d(\mathrm{Var}[Y|X=x])}{dP_{0\leftarrow 1}} = \frac{(e^\epsilon+1)^2\left(2(1-P_{0\leftarrow 0}) + |x|(2P_{0\leftarrow 0} - 1 - P_{0\leftarrow 1})\right)}{(1-P_{0\leftarrow 1})^3(e^\epsilon - 1)^2}. \qquad \text{(A.19)}$$

Then, we have following cases:

I. If $|x| = 0$, Eq. (A.19) $= \frac{(e^\epsilon+1)^2(2-2P_{0\leftarrow 0})}{(1-P_{0\leftarrow 1})^3(e^\epsilon-1)^2} > 0$,

II. If $|x| = 1$, Eq. (A.19) $= \frac{(e^\epsilon+1)^2}{(1-P_{0\leftarrow 1})^2(e^\epsilon-1)^2} > 0$.

Therefore, if given $P_{0\leftarrow 0}$, the variance of the output given input x is a strictly increasing function of $P_{0\leftarrow 1}$. Hence, we obtain the minimum variance when $P_{0\leftarrow 1} = \frac{P_{0\leftarrow 0}}{e^\epsilon}$.

■

A.4 Proof of Lemma 4

By summarizing Lemma 1, Lemma 2, and Lemma 3, our designed mechanism achieves the minimum variance when it satisfies $P_{0\leftarrow 1} = \frac{P_{0\leftarrow 0}}{e^\epsilon}$. Hence, the variance

is

$$\mathrm{Var}[Y|X=x] = C^2\,(1-P_{0\leftarrow 0}) + C^2 P_{0\leftarrow 0}(1-\frac{1}{e^\epsilon})|x| - x^2, \tag{A.20}$$

where

$$C = \frac{e^\epsilon+1}{(e^\epsilon-1)(1-\frac{P_{0\leftarrow 0}}{e^\epsilon})}. \tag{A.21}$$

For simplicity, we set

$$a = P_{0\leftarrow 0}, \tag{A.22}$$

$$\text{and } b = P_{0\leftarrow 0}(1-\frac{1}{e^\epsilon}) = a(1-\frac{1}{e^\epsilon}). \tag{A.23}$$

Since $x \in [-1,1]$, the worst-case noise variance is

$$\max_{x\in[-1,1]} \mathrm{Var}[Y|x] = \begin{cases} (1-a)C^2 + \frac{C^4b^2}{4}, & \text{if } \frac{C^2b}{2} < 1, \\ (1-a+b)C^2 - 1, & \text{if } \frac{C^2b}{2} \geq 1. \end{cases} \tag{A.24}$$

Substituting Eq. (A.21), Eq. (A.22), and Eq. (A.23) into Eq. (A.24) yields

$$\begin{aligned} &\max_{x\in[-1,1]} \mathrm{Var}[Y|x] \\ &= \begin{cases} \frac{(e^\epsilon+1)^2\cdot e^{2\epsilon}}{(e^\epsilon-1)^2}\left(\frac{1-a}{(e^\epsilon-a)^2} + \frac{(e^\epsilon+1)^2\cdot a^2}{4(e^\epsilon-a)^4}\right), & \text{if } \frac{C^2b}{2} < 1, \\ \frac{(e^\epsilon+1)^2\cdot e^{2\epsilon}}{(e^\epsilon-1)^2}\left(\frac{1-a}{(e^\epsilon-a)^2} + \frac{(e^\epsilon-1)\cdot a}{e^\epsilon(e^\epsilon-a)^2}\right) - 1, & \text{if } \frac{C^2b}{2} \geq 1. \end{cases} \\ &= \begin{cases} \frac{(e^\epsilon+1)^2\cdot e^{2\epsilon}}{(e^\epsilon-1)^2}\left(\frac{1-a}{(e^\epsilon-a)^2} + \frac{(e^\epsilon+1)^2\cdot a^2}{4(e^\epsilon-a)^4}\right), & \text{if } \frac{C^2b}{2} < 1, \\ \frac{(e^\epsilon+1)^2\cdot e^{\epsilon}}{(e^\epsilon-1)^2\cdot(e^\epsilon-a)} - 1, & \text{if } \frac{C^2b}{2} \geq 1. \end{cases} \end{aligned} \tag{A.25}$$

Substituting Eq. (A.23) and Eq. (A.21) into $\frac{C^2b}{2}$ yields

$$\begin{aligned} \frac{C^2b}{2} &= \frac{(e^\epsilon+1)^2\cdot e^{2\epsilon}}{2(e^\epsilon-1)^2(e^\epsilon-a)^2}\cdot\frac{a(e^\epsilon-1)}{e^\epsilon} \\ &= \frac{(e^\epsilon+1)^2\cdot e^\epsilon\cdot a}{2(e^\epsilon-1)(e^\epsilon-a)^2} \\ &< 1, \end{aligned} \tag{A.26}$$

which is equivalent to

$$2(e^\epsilon - 1)a^2 - \left[4(e^\epsilon - 1)e^\epsilon + (e^\epsilon + 1)^2 \cdot e^\epsilon\right] a + 2(e^\epsilon - 1)e^{2\epsilon} > 0. \tag{A.27}$$

In order to solve Eq. (A.27), we denote the smaller solution of the quadratic function as

$$a^* = \frac{e^\epsilon(e^{2\epsilon} + 6e^\epsilon - 3) - (e^\epsilon + 1)e^\epsilon\sqrt{(e^\epsilon + 1)^2 + 8(e^\epsilon - 1)}}{4(e^\epsilon - 1)}. \tag{A.28}$$

From Eq. (3.13), we get

$$P_{-C\leftarrow 0} \geq P_{-C\leftarrow 1}. \tag{A.29}$$

Then, substituting $P_{-C\leftarrow 0}$ and $P_{-C\leftarrow 1}$ with Eq. (A.12) and Eq. (A.14) in Eq. (A.29) yields

$$\frac{1 - P_{0\leftarrow 0}}{2} \geq \frac{1 - P_{0\leftarrow 1}}{e^\epsilon + 1}. \tag{A.30}$$

Hence, we obtain the value of a as follows:

$$a = P_{0\leftarrow 0} \leq \frac{e^\epsilon}{e^\epsilon + 2}. \tag{A.31}$$

From Eq. (A.31), we know that Eq. (A.27) will be ensured when (i) $0 \leq a < a^*$ if $a^* < \frac{e^\epsilon}{e^\epsilon+2}$, or (ii) $0 \leq a \leq \frac{e^\epsilon}{e^\epsilon+2}$ if $a^* \geq \frac{e^\epsilon}{e^\epsilon+2}$.

Hence, by combining with Eq. (A.25), we obtain

$$\max_{x\in[-1,1]} \mathrm{Var}[Y|x] = \begin{cases} \begin{cases} \frac{(e^\epsilon+1)^2 \cdot e^{2\epsilon}}{(e^\epsilon-1)^2}\left(\frac{1-a}{(e^\epsilon-a)^2} + \frac{(e^\epsilon+1)^2 \cdot a^2}{4(e^\epsilon-a)^4}\right), \\ \text{for } 0 \leq a < a^*, \end{cases} & \text{, if } a^* < \frac{e^\epsilon}{e^\epsilon+2} \\ \begin{cases} \frac{(e^\epsilon+1)^2 \cdot e^\epsilon}{(e^\epsilon-1)^2 \cdot (e^\epsilon-a)} - 1, \\ \text{for } a^* \leq a \leq \frac{e^\epsilon}{e^\epsilon+2}, \end{cases} & \\ \frac{(e^\epsilon+1)^2 \cdot e^{2\epsilon}}{(e^\epsilon-1)^2}\left(\frac{1-a}{(e^\epsilon-a)^2} + \frac{(e^\epsilon+1)^2 \cdot a^2}{4(e^\epsilon-a)^4}\right), & \\ \text{for } 0 \leq a \leq \frac{e^\epsilon}{e^\epsilon+2}, & \text{,if } a^* \geq \frac{e^\epsilon}{e^\epsilon+2}. \end{cases} \tag{A.32}$$

Substituting Eq. (A.28) into $a^* = \frac{e^\epsilon}{e^\epsilon+2}$ yields

$$\frac{e^\epsilon(e^{2\epsilon}+6e^\epsilon-3)-(e^\epsilon+1)e^\epsilon\sqrt{(e^\epsilon+1)^2+8(e^\epsilon-1)}}{4(e^\epsilon-1)} = \frac{e^\epsilon}{e^\epsilon+2}. \tag{A.33}$$

After solving Eq. (A.33), we get $\epsilon = \ln 4$. According to Fig. A.1, we obtain that

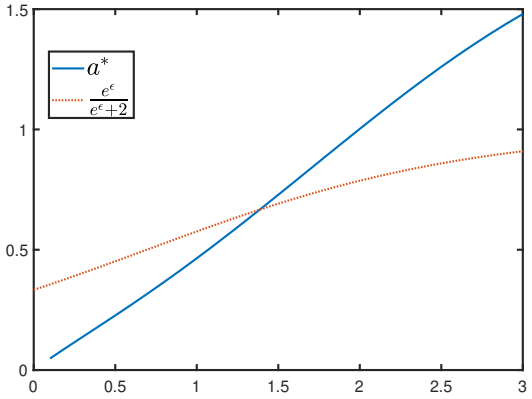


FIGURE A.1: Compare a^* with $\frac{e^\epsilon}{e^\epsilon+2}$.

$a^* \geq \frac{e^\epsilon}{e^\epsilon+2}$ if $0 < \epsilon \leq \ln 4$. Since $\epsilon = \ln 4$ is the only solution if $\epsilon > 0$, we conclude that $a^* > \frac{e^\epsilon}{e^\epsilon+2}$ if $\epsilon > \ln 4$. Therefore, we can replace the condition $a^* < \frac{e^\epsilon}{e^\epsilon+2}$ and write the variance as follows:

$\max_{x\in[-1,1]} \mathrm{Var}[Y|x] =$

$$\begin{cases} \begin{cases} \frac{(e^\epsilon+1)^2\cdot e^{2\epsilon}}{(e^\epsilon-1)^2}\left(\frac{1-a}{(e^\epsilon-a)^2}+\frac{(e^\epsilon+1)^2\cdot a^2}{4(e^\epsilon-a)^4}\right), \\ \text{for } 0 \leq a < a^*, \end{cases} & \text{, if } \epsilon < \ln 4, \qquad \text{(A.34a)} \\ \begin{cases} \frac{(e^\epsilon+1)^2\cdot e^\epsilon}{(e^\epsilon-1)^2\cdot(e^\epsilon-a)} - 1, \\ \text{for } a^* \leq a \leq \frac{e^\epsilon}{e^\epsilon+2}, \end{cases} & \end{cases}$$

$$\begin{cases} \frac{(e^\epsilon+1)^2\cdot e^{2\epsilon}}{(e^\epsilon-1)^2}\left(\frac{1-a}{(e^\epsilon-a)^2}+\frac{(e^\epsilon+1)^2\cdot a^2}{4(e^\epsilon-a)^4}\right), & \text{(A.34b)} \\ \text{for } 0 \leq a \leq \frac{e^\epsilon}{e^\epsilon+2}, \qquad \text{if } \epsilon \geq \ln 4. \end{cases}$$

To simplify the calculation of $\max_{x\in[-1,1]} \mathrm{Var}[Y|x]$ in Eq. (A.34), we define

$$f_1(a) := \frac{(e^\epsilon+1)^2 \cdot e^{2\epsilon}}{(e^\epsilon-1)^2}\left(\frac{1-a}{(e^\epsilon-a)^2} + \frac{(e^\epsilon+1)^2 \cdot a^2}{4(e^\epsilon-a)^4}\right), \tag{A.35}$$

and

$$f_2(a) := \frac{(e^\epsilon+1)^2 \cdot e^\epsilon}{(e^\epsilon-1)^2 \cdot (e^\epsilon-a)} - 1. \tag{A.36}$$

The first-order derivative of $f_2(a)$ in Eq. (A.36) is

$$f_2'(a) = \frac{e^\epsilon(e^\epsilon+1)^2}{(e^\epsilon-1)^2(e^\epsilon-a)^2} > 0. \tag{A.37}$$

As $f_2'(a) > 0$, the worst-case noise variance monotonously increases when $a \in [a^*, \frac{e^\epsilon}{e^\epsilon+2}]$. Then, we can get the optimal a by analyzing $f_1(a)$ in Eq. (A.35) when $a \in [0, a^*)$ if $\epsilon < \ln 4$.

The first-order derivative of Eq. (A.35) is

$$f_1'(a) = \frac{2(1-a)}{(e^\epsilon-a)^3} - \frac{1}{(e^\epsilon-a)^2} + \frac{a(e^\epsilon+1)^2}{2(e^\epsilon-a)^4} + \frac{a^2(e^\epsilon+1)^2}{(e^\epsilon-a)^5}. \tag{A.38}$$

After simplifying $f_1'(a)$, we have

$$f_1'(a) = \frac{-2a^3 - a^2(-e^{2\epsilon}-5-4e^\epsilon)}{2(e^\epsilon-a)^5} + \frac{-a(7e^\epsilon-4e^{2\epsilon}-e^{3\epsilon}) - (2e^{3\epsilon}-4e^{2\epsilon})}{2(e^\epsilon-a)^5}. \tag{A.39}$$

Since $2(e^\epsilon-a)^5 > 0$, solving $f_1'(a) = 0$ is equivalent to solve the following equation

$$2a^3 + a^2(-e^{2\epsilon}-5-4e^\epsilon) + a(7e^\epsilon-4e^{2\epsilon}-e^{3\epsilon}) + (2e^{3\epsilon}-4e^{2\epsilon}) = 0. \tag{A.40}$$

We define coefficients of Eq. (A.40) as follows:

$$c_3 := 2, \tag{A.41}$$

$$c_2 := -e^{2\epsilon}-5-4e^\epsilon, \tag{A.42}$$

$$c_1 := 7e^\epsilon-4e^{2\epsilon}-e^{3\epsilon}, \tag{A.43}$$

$$\text{and } c_0 := 2e^{3\epsilon}-4e^{2\epsilon}. \tag{A.44}$$

The general solution of the cubic equation involves calculation of

$$\begin{aligned}
\Delta_0 &= {c_2}^2 - 3c_3c_1 \\
&= (-e^{2\epsilon} - 5 - 4e^{\epsilon})^2 - 3 \times 2(7e^{\epsilon} - 4e^{2\epsilon} - e^{3\epsilon}) \\
&= e^{4\epsilon} + 14e^{3\epsilon} + 50e^{2\epsilon} - 2e^{\epsilon} + 25 > 0, \\
\Delta_1 &= 2{c_2}^3 - 9c_3c_2c_1 + 27c_3^2c_0 \\
&= 2(-e^{2\epsilon} - 5 - 4e^{\epsilon})^3 \\
&\quad - 9 \times 2(-e^{2\epsilon} - 5 - 4e^{\epsilon})(7e^{\epsilon} - 4e^{2\epsilon} - e^{3\epsilon}) \\
&\quad + 27 \times 2^2(2e^{3\epsilon} - 4e^{2\epsilon}) \\
&= -2e^{6\epsilon} - 42e^{5\epsilon} - 270e^{4\epsilon} - 404e^{3\epsilon} - 918e^{2\epsilon} \\
&\quad + 30e^{\epsilon} - 250 < 0, \\
\text{and } I &= \sqrt[3]{\frac{\Delta_1 \pm \sqrt{\Delta_1^2 - 4\Delta_0^3}}{2}}.
\end{aligned}$$

Substituting Δ_0 and Δ_1 into $\Delta_1^2 - 4\Delta_0^3$ yields

$$\begin{aligned}
&\Delta_1^2 - 4\Delta_0^3 \\
&\quad = (-2c^6 - 42c^5 - 270c^4 - 404c^3 - 918c^2 + 30c - 250)^2 \\
&\quad - 4(c^4 + 14c^3 + 50c^2 - 2c + 25)^3 < 0,
\end{aligned}$$

and then we obtain

$$\sqrt{\Delta_1^2 - 4\Delta_0^3} = i\sqrt{4\Delta_0^3 - \Delta_1^2}. \tag{A.45}$$

Finally, we pick

$$I = \sqrt[3]{\frac{\Delta_1 - i\sqrt{4\Delta_0^3 - \Delta_1^2}}{2}}.$$

To eliminate the imaginary number, we change Eq. (A.131) using Euler's formula. Then, we have

$$|I| = (|I|^3)^{1/3} = \left(\sqrt{\frac{\Delta_1^2}{4} + \Delta_0^3 - \frac{\Delta_1^2}{4}}\right)^{1/3} = \sqrt{\Delta_0}, \tag{A.46}$$

$$I = |I|e^{i\theta}, \tag{A.47}$$

$$\text{and } I^3 = |I|^3 e^{3i\theta} = \sqrt{\Delta_0^3} e^{3i\theta}. \tag{A.48}$$

Therefore, we obtain $I = \sqrt{\Delta_0} e^{i\theta}$.

According to Euler's formula, we have

$$e^{i3\theta} = \cos 3\theta + i \sin 3\theta, \tag{A.49}$$

$$\cos 3\theta = \frac{\Delta_1}{2\Delta_0^{\frac{3}{2}}} < 0, \tag{A.50}$$

$$\text{and } \sin 3\theta = -\frac{\sqrt{4\Delta_0^3 - \Delta_1^2}}{2\Delta_0^{\frac{3}{2}}} < 0. \tag{A.51}$$

Hereby, we obtain

$$3\theta = -\pi + \arccos(-\frac{\Delta_1}{2\Delta_0^{\frac{3}{2}}}) \tag{A.52}$$

$$\text{and } \theta = -\frac{\pi}{3} + \frac{1}{3}\arccos(-\frac{\Delta_1}{2\Delta_0^{\frac{3}{2}}}). \tag{A.53}$$

The solution of the cubic function is

$$a_k = -\frac{1}{3c_3}(c_2 + \xi^k I + \frac{\Delta_0}{\xi^k I}),\ k \in \{0, 1, 2\},\ \text{where } \xi = \frac{-1 + \sqrt{-3}}{2}. \tag{A.54}$$

To solve Eq. (A.54), we have following cases:

- If $k = 0$, we have

$$a_0 = -\frac{1}{3c_3}(c_2 + I + \frac{\Delta_0}{I}). \tag{A.55}$$

  Substituting θ (A.53) and c_2 (A.42) into Eq. (A.55) yields

$$a_0 = -\frac{1}{6}(-e^{2\epsilon} - 4e^{\epsilon} - 5 + 2\sqrt{\Delta_0}\cos(-\frac{\pi}{3} + \frac{1}{3}\arccos(-\frac{\Delta_1}{2\Delta_0^{\frac{3}{2}}}))). \tag{A.56}$$

- If $k = 1$, we have

$$\begin{aligned} a_1 &= -\frac{1}{3c_3}(c_2 + \xi I + \frac{\Delta_0}{\xi I}) \\ &= -\frac{1}{3c_3}(c_2 + (-\frac{1}{2} + \frac{\sqrt{3}i}{2})I + \frac{\Delta_0}{(-\frac{1}{2} + \frac{\sqrt{3}i}{2})I}) \\ &= -\frac{1}{3c_3}(c_2 + Ie^{i\frac{2\pi}{3}} + \frac{\Delta_0}{I}e^{i\frac{4\pi}{3}}) \\ &= -\frac{1}{3c_3}(c_2 + \sqrt{\Delta_0}e^{i(\theta+\frac{2}{3}\pi)} + \sqrt{\Delta_0}e^{i(\frac{4\pi}{3}-\theta)}). \end{aligned} \tag{A.57}$$

  After simplifying Eq. (A.57) using Eq. (A.49), we have

$$a_1 = -\frac{1}{3c_3}(c_2 + 2\sqrt{\Delta_0}\cos(\theta + \frac{2\pi}{3})). \tag{A.58}$$

  Substituting θ (A.53) and c_2 (A.42) into Eq. (A.58) yields

$$a_1 = -\frac{1}{6}(-e^{2\epsilon} - 4e^{\epsilon} - 5 + 2\sqrt{\Delta_0}\cos(\frac{\pi}{3} + \frac{1}{3}\arccos(-\frac{\Delta_1}{2\Delta_0^{\frac{3}{2}}}))). \tag{A.59}$$

- If $k = 2$, we get

$$\begin{aligned} a_2 &= -\frac{1}{3c_3}(c_2 + \xi^2 I + \frac{\Delta_0}{\xi^2 I}) \\ &= -\frac{1}{3c_3}(c_2 + (-\frac{1}{2} + \frac{\sqrt{3}i}{2})^2 I + \frac{\Delta_0}{(-\frac{1}{2} + \frac{\sqrt{3}i}{2})^2 I}) \\ &= -\frac{1}{3c_3}(c_2 + (-\frac{1}{2} - \frac{\sqrt{3}i}{2})I + \frac{\Delta_0}{(-\frac{1}{2} - \frac{\sqrt{3}i}{2})I}) \\ &= -\frac{1}{3c_3}(c_2 + \sqrt{\Delta_0}e^{i(\theta+\frac{4\pi}{3})} + \sqrt{\Delta_0}e^{i(\frac{2\pi}{3}-\theta)}). \end{aligned} \tag{A.60}$$

  Simplify Eq. (A.60) using Eq. (A.49), and then we obtain

$$a_2 = -\frac{1}{3c_3}(c_2 + 2\sqrt{\Delta_0}\cos(\theta - \frac{2\pi}{3})). \tag{A.61}$$

  Substituting θ (A.53) and c_2 (A.42) into Eq. (A.61) yields

$$a_2 = -\frac{1}{6}(-e^{2\epsilon} - 4e^{\epsilon} - 5 + 2\sqrt{\Delta_0}\cos(-\pi + \frac{1}{3}\arccos(-\frac{\Delta_1}{2\Delta_0^{\frac{3}{2}}}))). \tag{A.62}$$

The discriminant of the cubic equation determines the number of real and complex roots as follows:

$$\Delta = 18c_3c_2c_1c_0 - 4c_2^3c_0 + c_2^2c_1^2 - 4c_3c_1^3 - 27c_3^2c_0^2. \tag{A.63}$$

Substituting c_3 (A.41), c_2 (A.42), c_1 (A.43) and c_0 (A.44) into Δ (A.63) yields

$$\Delta = e^{2\epsilon}(e^{\epsilon}+1)^2(e^{6\epsilon} + 30e^{5\epsilon} + 279e^{4\epsilon} + 580e^{3\epsilon} - 2385e^{2\epsilon} + 606e^{\epsilon} - 775). \tag{A.64}$$

If $\Delta = 0$, we get $\epsilon \approx 0.629598$.

- If $0 < \epsilon < 0.629598,\ \Delta < 0$, and the equation has one real root and two non-real complex conjugate roots.
- If $\epsilon = 0.629598,\ \Delta = 0$, and the equation has a real multiple root.
- If $\epsilon > 0.629598,\ \Delta > 0$, and the equation has three distinct real roots.

From the simplified $f_1'(a)$ in Eq. (A.39), we obtain its numerator as follows:

$$g(a) := -2a^3 - a^2(-e^{2\epsilon} - 5 - 4e^{\epsilon}) - a(7e^{\epsilon} - 4e^{2\epsilon} - e^{3\epsilon}) - (2e^{3\epsilon} - 4e^{2\epsilon}). \tag{A.65}$$

Let $c = e^{\epsilon}$, and then we change $g(a)$ to the following:

$$g(a) = -2a^3 - a^2(-c^2 - 5 - 4c) - a(7c - 4c^2 - c^3) - (2c^3 - 4c^2). \tag{A.66}$$

Case 1: If $0 < \epsilon < 0.629598$, $g(a)$ has one real root and two non-real complex conjugate roots. As a_0 and a_1 are conjugate, a_2 is the real root. Since $g(0) = -(2c^3 - 4c^2) > 0$ and Fig. A.2 shows that $a_2 > 1$, we conclude that $g(a) > 0$ when $a \in [0, 1]$. Thus, $f_1(a)$ monotonously increases. Therefore, we obtain the minimum $f_1(a)$ at $a = 0$.

Case 2: If $0.629598 \leq \epsilon < \ln 2$, $\Delta \geq 0$, so we get real roots. Hence, we have following cases:

- If $a = 0,\ g(0) = -(2c^3 - 4c^2) > 0$.
- If $a = 2,\ g(2) = 2(8c^2 + c + 2) > 0$.

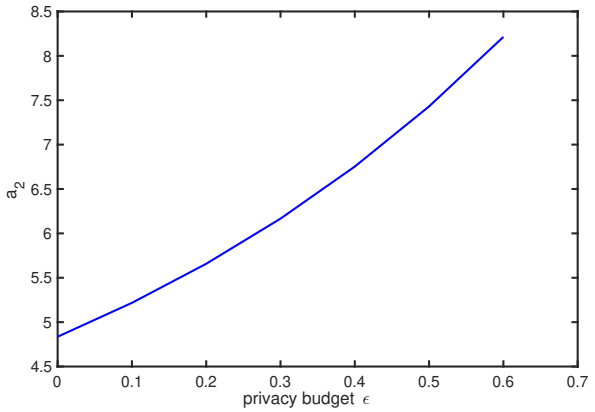


FIGURE A.2: $g(a)$ if $\epsilon \in [0, 0.629598]$.

- If $a = +\infty$, $\lim_{a\to\infty} g(a) = -\infty < 0$.

Since $g(2) > 0$ and $\lim_{a\to\infty} g(a) < 0$, we have a root in $(2, +\infty)$. Based on the properties of cubic function, we have

$$a_0a_1 + a_0a_2 + a_1a_2 = \frac{c_1}{c_3} = \frac{7e^{\epsilon} - 4e^{2\epsilon} - e^{3\epsilon}}{2}, \tag{A.67}$$

$$\text{and } a_0a_1a_2 = -\frac{c_0}{c_3} = -\frac{2e^{3\epsilon} - 4e^{2\epsilon}}{2}. \tag{A.68}$$

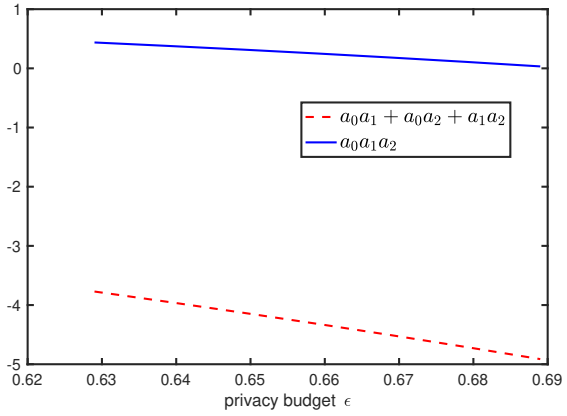


FIGURE A.3: $a_0a_1 + a_0a_2 + a_1a_2$ and $a_0a_1a_2$ if $\epsilon \in [0.629598, \ln 2]$.

Fig. A.3 shows that $a_0a_1+a_0a_2+a_1a_2 < 0$ (Eq. (A.67)) and $a_0a_1a_2 > 0$ (Eq. (A.68)). From $a_0a_1a_2 > 0$, we conclude that there are one positive root and two negative roots or three positive roots, and then we conclude that there are one positive root and two negative roots because $a_0a_1+a_0a_2+a_1a_2 < 0$. Since two negative roots are

out of the a's domain, we only discuss the positive root. Thus, we have following cases:

- If $a \in [0, root)$, $g(a) > 0$ meaning $f_1'(a) > 0$.
- If $a \in [root, +\infty)$, $g(a) \leq 0$ meaning $f_1'(a) \leq 0$.

From the above, as the positive real root is in $(2, +\infty)$, $g(a) > 0$ and $f_1(a)$ monotonously increases when $a \in [0, \frac{e^\epsilon}{e^\epsilon+2}]$. Therefore, we obtain the minimum $f_1(a)$ when $a = 0$.

Case 3: If $\ln 2 \leq \epsilon \leq \ln 5.53$, $\Delta > 0$, so there are three distinct real roots. As $a_0 a_1 a_2 < 0$, there are one negative root and two positive roots or three negative roots. Only one negative root and two positive roots satisfy $a_0 a_1 + a_0 a_2 + a_1 a_2 < 0$. Thus, we have following cases:

- If $a = 0$, $g(0) = -(2c^3 - 4c^2) < 0$.
- If $a = 2$, $g(2) = 2(8c^2 + c + 2) > 0$.
- If $a = +\infty$, $\lim_{a\to\infty} g(a) = -\infty$.

From above results, we can deduce that there is one positive root in $(0, 2)$ defined as $root_1$, and the other positive root is in $(2, +\infty)$ defined as $root_2$. Since $root_2 > 1$ is out of a's domain, we only discuss $root_1$. Thus, we have following cases:

- If $a \in [0, root_1]$, $g(a) \leq 0$.
- If $a \in (root_1, root_2)$, $g(a) > 0$.

Therefore, if $g(\frac{c}{c+2}) \geq 0$, we conclude that $root_1 \leq \frac{c}{c+2}$. The exact form of $g(\frac{c}{c+2})$ is

$$g\left(\frac{c}{c+2}\right) = \frac{c^2(c+1)^2(-c^2+3c+14)}{(c+2)^3}. \tag{A.69}$$

By solving $g(\frac{c}{c+2}) \geq 0$, we have $c \leq \ln\left(\frac{3+\sqrt{65}}{2}\right) \approx 5.53$, i.e., $\epsilon \leq \ln 5.53$. From Fig. A.4, we can conclude that a_1 is the correct root, $a_0 < 0$ and $a_2 > 1$, and we get $a = a_1 = -\frac{1}{6}(-e^{2\epsilon} - 4e^\epsilon - 5 + 2\sqrt{\Delta_0}\cos(\frac{\pi}{3} + \frac{1}{3}\arccos(-\frac{\Delta_1}{2\Delta_0^{\frac{3}{2}}})))$.

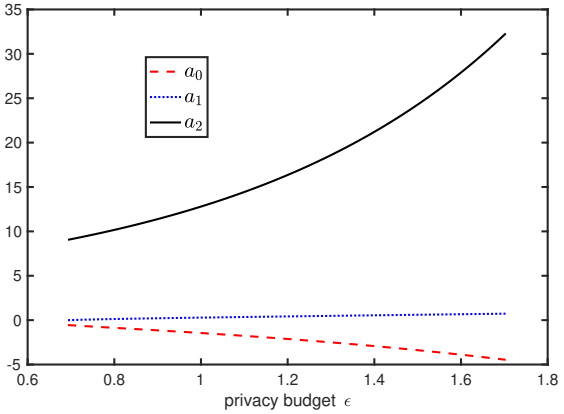


FIGURE A.4: a_0, a_1 and a_2 if $\epsilon \in [\ln 2, \ln 5.53]$.

Case 4: If $\epsilon > \ln 5.53$, $\Delta > 0$, so there are three distinct real roots.

From the analysis in Case 3, we know that if $\epsilon > \ln 5.53$, $root_1 > \frac{c}{c+2}$ and $g(\frac{c}{c+2}) < 0$. We know that $g(a) \leq 0$ if $a \in [0, \frac{c}{c+2}]$, meaning $f_1'(a) < 0$, so that $f_1(a)$ monotonously decreases if $\epsilon > \ln 5.53$. Therefore, we obtain the minimum $f_1(a)$ when $a = \frac{e^\epsilon}{e^\epsilon+2}$.

By summarizing the above, we obtain the optimal a which is named as $P_{0\leftarrow 0}$ in the Eq. (3.4). ■

A.5 Proof of Lemma 5

By substituting the optimal $P_{0\leftarrow 0}$ of Eq. (3.4) with a in the $\max_{x\in[-1,1]} \mathrm{Var}[Y|x]$ of Eq. (A.34), we obtain the worst-case noise variance of `Three-Outputs` as follows:

$$\min_{P_{0\leftarrow 0}} \max_{x\in[-1,1]} \mathrm{Var}[Y|x] =$$

$$\begin{cases} \frac{(e^\epsilon+1)^2}{(e^\epsilon-1)^2}, \text{ for } \epsilon < \ln 2, \\ \frac{(e^\epsilon+1)^2 \cdot e^{2\epsilon}}{(e^\epsilon-1)^2}\left(\frac{1-P_{0\leftarrow 0}}{(e^\epsilon-P_{0\leftarrow 0})^2} + \frac{(e^\epsilon+1)^2 \cdot P_{0\leftarrow 0}^2}{4(e^\epsilon-P_{0\leftarrow 0})^4}\right), \text{ for } \ln 2 \leq \epsilon \leq \ln 5.53, \\ \text{where } P_{0\leftarrow 0} = -\frac{1}{6}(-e^{2\epsilon} - 4e^\epsilon - 5 \\ \quad +2\sqrt{\Delta_0}\cos(\frac{\pi}{3} + \frac{1}{3}\arccos(-\frac{\Delta_1}{2\Delta_0^{\frac{3}{2}}}))), \\ \frac{(e^\epsilon+2)(e^\epsilon+10)}{4(e^\epsilon-1)^2}, \quad \text{for } \epsilon > \ln 5.53. \end{cases} \tag{A.70}$$

■

A.6 Proving Lemma 6

From Eq. (3.25a) and Eq. (3.25b), for any $Y \in [-A, A]$ and any two input values $x_1, x_2 \in [-1, 1]$, we have $\frac{\mathbb{F}[Y|x_1]}{\mathbb{F}[Y|x_2]} \leq \frac{c}{d} = \exp(\epsilon)$. Thus, Algorithm 3 satisfies local differential privacy. For notational simplicity, with a fixed ϵ below, we will write $L(\epsilon, x, t)$ and $R(\epsilon, x, t)$ as L_x and R_x. Based on the proper distribution, we have

$$\int_{-A}^{A} \mathbb{F}\left[Y = y|x\right] \, \mathrm{d}y = c(R_x - L_x) + d[2A - (R_x - L_x)] = 1. \tag{A.71}$$

In addition, we prove the unbiased estimation as follows:

$$\begin{aligned} \mathbb{E}[Y = y|x] &= \int_{-A}^{A} y \cdot \mathbb{F}\left[Y = y|x\right] \, \mathrm{d}y \\ &= \frac{d}{2} \cdot ({L_x}^2 - A^2) + \frac{c}{2}({R_x}^2 - {L_x}^2) + \frac{d}{2} \cdot (A^2 - {R_x}^2) \\ &= x. \end{aligned} \tag{A.72}$$

By solving above Eq. (A.71) and Eq. (A.72), we have

$$\begin{cases} L_x &= \frac{x}{1-2Ad} - \frac{1-2Ad}{2(c-d)}, \\ R_x &= \frac{x}{1-2Ad} + \frac{1-2Ad}{2(c-d)}. \end{cases} \tag{A.73}$$

With $-A \leq y \leq A$, the constraint $-A \leq L_x < R_x \leq A$ for any $-1 \leq x \leq 1$ in Eq. (3.25), Eq. (3.27), and Eq. (3.28) implies

$$Ad < \frac{1}{2}, \tag{A.74}$$

$$\text{and } A \geq \frac{1}{1-2Ad} + \frac{1-2Ad}{2(c-d)}. \tag{A.75}$$

For notational simplicity, we define α and ξ as

$$\alpha := Ad, \tag{A.76}$$

$$\text{and } \xi := \frac{c-d}{d}, \tag{A.77}$$

where it is clear under privacy parameter ϵ that

$$\xi = e^{\epsilon} - 1. \tag{A.78}$$

Applying Eq. (A.76) and Eq. (A.77) to Inequality (A.74) and Eq. (A.75), we obtain

$$\frac{\alpha}{d} \geq \frac{1}{1-2\alpha} + \frac{1-2\alpha}{2\xi d}, \tag{A.79}$$

$$\text{and } \alpha < \frac{1}{2}. \tag{A.80}$$

The condition Eq. (A.79) induces $d \leq \frac{(2\xi+4)\alpha-(4+4\xi)\alpha^2-1}{2\xi} = \frac{[(2\xi+2)\alpha-1](1-2\alpha)}{2\xi}$. In view of $\frac{1}{2(\xi+1)} = \frac{1}{2e^\epsilon} < \alpha < \frac{1}{2}$, we define t satisfying $0 < t < \infty$ such that $\alpha = \frac{t+1}{2(t+e^\epsilon)}$. Note that $\lim_{t\to 0} \frac{t+1}{2(t+e^\epsilon)} = \frac{1}{2e^\epsilon}$, $\lim_{t\to\infty} \frac{t+1}{2(t+e^\epsilon)} = \frac{1}{2}$ and $d = \frac{[(2\xi+2)\alpha-1](1-2\alpha)}{2\xi} = \frac{1}{2\xi} \cdot \frac{\xi t}{t+1+\xi} \cdot \frac{\xi}{t+1+\xi} = \frac{t(e^\epsilon-1)}{2(t+e^\epsilon)^2}$.

By applying α, d and ξ to Eq. (A.73), we have

$$\begin{cases} L_x &= \frac{x}{1-2\alpha} - \frac{1-2\alpha}{2\xi d} = x \cdot \frac{e^\epsilon+t}{e^\epsilon-1} - \frac{e^\epsilon+t}{t(e^\epsilon-1)} = \frac{(e^\epsilon+t)(xt-1)}{t(e^\epsilon-1)}, \\ R_x &= \frac{x}{1-2\alpha} + \frac{1-2\alpha}{2\xi d} = x \cdot \frac{e^\epsilon+t}{e^\epsilon-1} + \frac{e^\epsilon+t}{t(e^\epsilon-1)} = \frac{(e^\epsilon+t)(xt+1)}{t(e^\epsilon-1)}. \end{cases} \tag{A.81}$$

Furthermore, the variance of Y is

$$\begin{aligned} \mathrm{Var}[Y|x] &= \mathbb{E}[Y^2|\ x] - (\mathbb{E}[Y|x])^2 \\ &= \int_{-A}^{A} y^2 \mathbb{F}\left[Y = y|x\right]\ \mathrm{d}y - x^2 \\ &= \int_{-A}^{L_x} dy^2\, \mathrm{d}y + \int_{L_x}^{R_x} cy^2\, \mathrm{d}y + \int_{R_x}^{A} dy^2\, \mathrm{d}y - x^2 \\ &= \frac{d}{3}[{L_x}^3 - (-A)^3] + \frac{c}{3}({R_x}^3 - {L_x}^3) + \frac{d}{3}(A^3 - {R_x}^3) - x^2 \\ &= \frac{2d}{3}A^3 + \frac{(c-d)}{3}({R_x}^3 - {L_x}^3) - x^2. \end{aligned} \tag{A.82}$$

Substituting Eq. (A.73) into Eq. (A.82) yields

$$\begin{aligned}\mathrm{Var}[Y|x] &= \frac{2d}{3}A^3 + \frac{(c-d)}{3}\cdot \\ &\quad \left[\left(\frac{x}{1-2Ad} + \frac{1-2Ad}{2(c-d)}\right)^3 - \left(\frac{x}{1-2Ad} - \frac{1-2Ad}{2(c-d)}\right)^3\right] \\ &\quad - x^2 \\ &= \frac{2d}{3}A^3 + \frac{(c-d)}{3}\cdot \\ &\quad \left\{6\left(\frac{x}{1-2Ad}\right)^2 \times \frac{1-2Ad}{2(c-d)} + 2\left[\frac{1-2Ad}{2(c-d)}\right]^3\right\} - x^2 \\ &= \left(\frac{1}{1-2Ad} - 1\right)x^2 + \frac{2d}{3}A^3 + \frac{(1-2Ad)^3}{12(c-d)^2}. \end{aligned} \tag{A.83}$$

Substituting $1-2\alpha = \frac{e^\epsilon-1}{e^\epsilon+t}$, $d = \frac{t(e^\epsilon-1)}{2(t+e^\epsilon)^2}$, $\xi = \frac{c-d}{d} = e^\epsilon - 1$, and $\alpha = \frac{t+1}{2(t+e^\epsilon)}$ into Eq. (A.83) yields

$$\mathrm{Var}[Y|x] = \frac{t+1}{e^\epsilon-1}x^2 + \frac{(t+e^\epsilon)\big((t+1)^3+e^\epsilon-1\big)}{3t^2(e^\epsilon-1)^2}. \tag{A.84}$$

■

A.7 Calculation of Value t

In order to find the optimal t for $\min_t \max_{x\in[-1,1]} \mathrm{Var}[Y|x]$, we calculate the first-order derivative of $\max_{x\in[-1,1]} \mathrm{Var}[Y|x]$ as follows:

$$\begin{aligned}&\frac{2t}{3(e^\epsilon-1)^2} + \frac{4}{3(e^\epsilon-1)} + \frac{4}{3(e^\epsilon-1)^2} - \frac{4t^{-2}}{3(e^\epsilon-1)^2} \\ &- \frac{4t^{-2}}{3(e^\epsilon-1)} - \frac{2t^{-3}}{3(e^\epsilon-1)^2} - \frac{4t^{-3}}{3(e^\epsilon-1)} - \frac{2t^{-3}}{3} \\ &= \frac{2}{3(e^\epsilon-1)^2}[t + 2e^\epsilon - 2e^\epsilon t^{-2} - e^{2\epsilon}t^{-3}]. \end{aligned} \tag{A.85}$$

Next, we calculate the second-order derivative of $\max_{x\in[-1,1]} \mathrm{Var}[Y|x]$ as follows:

$$\frac{2}{3(e^\epsilon-1)^2}[1 + 4e^\epsilon t^{-3} + 3e^\epsilon t^{-4}] > 0. \tag{A.86}$$

Since the second-order derivative of $\max_{x\in[-1,1]} \mathrm{Var}[Y|x] > 0$, we conclude that $\max_{x\in[-1,1]} \mathrm{Var}[Y|x]$ has the minimum point in its domain.

To find t which minimizes $\max_{x\in[-1,1]} \mathrm{Var}[Y|x]$, we set $t^4 + 2e^{\epsilon}t^3 - 2e^{\epsilon}t - e^{2\epsilon} = 0$. By solving

$$t^4 + 2e^{\epsilon}t^3 - 2e^{\epsilon}t - e^{2\epsilon} = 0, \tag{A.87}$$

we obtain Eq. (3.29).

Define Eq. (A.87)'s coefficients as $c_4 := 1,\ c_3 := 2e^{\epsilon},\ c_2 := 0,\ c_1 := -2e^{\epsilon}$, and $c_0 := -e^{2\epsilon}$, and then we obtain

$$c_4 \cdot t^4 + c_3 \cdot t^3 + c_1 \cdot t + c_0 = 0. \tag{A.88}$$

To change Eq. (A.88) into a depressed quartic form, we substitute $f := e^{\epsilon}, t := y - \frac{c_3}{4c_4} = y - \frac{f}{2}$ into Eq. (A.88) and obtain

$$y^4 + p \cdot y^2 + q \cdot y + r = 0, \tag{A.89}$$

where

$$p = \frac{8c_2c_4 - 3c_3^2}{8c_4^2} = -\frac{3f^2}{2}, \tag{A.90}$$

$$q = \frac{c_3^3 - 4c_2c_3c_4 + 8c_1c_4^2}{8c_4^3} = f^3 - 2f, \tag{A.91}$$

$$\text{and } r = \frac{-3c_3^4 + 256c_0c_4^3 - 64c_1c_3c_4^2 + 16c_2c_3^2c_4}{256c_4^4} = -\frac{3}{16}f^4. \tag{A.92}$$

Rewrite Eq. (A.89) to the following

$$\left(y^2 + \frac{p}{2}\right) = -qy - r + \frac{p^2}{4}. \tag{A.93}$$

Then, we introduce a variable m into the factor on the left-hand side of Eq. (A.93) by adding $2y^2m + pm + m^2$ to both sides. Thus, we can change the equation to the following:

$$\left(y^2 + \frac{p}{2} + m\right) = 2my^2 - qy + m^2 + mp + \frac{p^2}{4} - r. \tag{A.94}$$

Since m is arbitrarily chosen, we choose the value of m to get a perfect square in the right-hand side. Hence, we obtain that

$$8m^3 + 8pm^2 + (2p^2 - 8r)m - q^2 = 0. \tag{A.95}$$

In order to solve Eq. (A.95), we substitute Eq. (A.90), Eq. (A.91), and Eq. (A.92) into the following equations:

$$\begin{aligned}
c_3' &:= 8, \\
c_2' &:= 8p, \\
c_1' &:= 2p^2 - 8r, \\
c_0' &:= -q^2, \\
\Delta_0' &= (c_2')^2 - 3c_3'c_1' = (8p)^2 - 3 \cdot 8 \cdot (2p^2 - 8r) = 0, \\
\Delta_1' &= 2(c_2')^3 - 9c_3' \cdot c_2' \cdot c_1' + 27(c_3')^2 \cdot c_0' \\
&= 2(8p)^3 - 9 \cdot 8 \cdot 8p \cdot (2p^2 - 8r) + 27 \cdot 8^2 \cdot (-q^2) \\
&= 6912(f^4 - f^2), \\
\text{and } I' &= \sqrt[3]{\frac{\Delta_1' \pm \sqrt{(\Delta_1')^2 - 4(\Delta_0')^3}}{2}}.
\end{aligned}$$

As

$$I' = \sqrt[3]{\frac{\Delta_1' - \sqrt{(\Delta_1')^2 - 4(\Delta_0')^3}}{2}} = 0 \text{ cannot be used as the denominator,}$$

we take

$$\begin{aligned}
I' &= \sqrt[3]{\frac{\Delta_1' + \sqrt{(\Delta_1')^2 - 4(\Delta_0')^3}}{2}} = \sqrt[3]{\Delta_1'} \\
&= \sqrt[3]{6912(f^4 - f^2)}.
\end{aligned} \tag{A.96}$$

By solving the cubic function in Eq. (A.96), we have roots as follows:

$$\begin{aligned} m_k &= -\frac{1}{3c_3'}(c_2' + \xi^k I' + \frac{\Delta_0'}{\xi^k I'}) \\ &= \frac{f^2}{2} + \sqrt[3]{\frac{f^2 - f^4}{2}}\xi^k, \ k \in \{0, 1, 2\}, \\ \text{where } \xi &= \frac{-1 + \sqrt{-3}}{2}. \end{aligned} \tag{A.97}$$

We only use the real-value root; thus, we get

$$m = \frac{f^2}{2} + \sqrt[3]{\frac{f^2 - f^4}{2}}. \tag{A.98}$$

Thus, we obtain

$$y = \frac{\pm_1\sqrt{2m} \pm_2 \sqrt{-(2p + 2m \pm_1 \frac{\sqrt{2}q}{\sqrt{m}})}}{2}. \tag{A.99}$$

Then, the solutions of the original quartic equation are

$$t = -\frac{e^\epsilon}{2} + \frac{\pm_1\sqrt{2m} \pm_2 \sqrt{-(2p + 2m \pm_1 \frac{\sqrt{2}q}{\sqrt{m}})}}{2}. \tag{A.100}$$

After substituting m, p, q, f into Eq. (A.100), we obtain t as follows:

$$
t = \begin{cases}
\frac{1}{2}\sqrt{e^{2\epsilon}+2^{2/3}\sqrt[3]{e^{2\epsilon}-e^{4\epsilon}}}+ & \\
\frac{1}{2}\sqrt{2e^{2\epsilon}-2^{2/3}\sqrt[3]{e^{2\epsilon}-e^{4\epsilon}}+\frac{4e^{\epsilon}-2e^{3\epsilon}}{\sqrt{e^{2\epsilon}+2^{2/3}\sqrt[3]{e^{2\epsilon}-e^{4\epsilon}}}}} & \\
-\frac{e^{\epsilon}}{2} > 0, \quad \text{if } \epsilon < \ln\sqrt{2}, & \text{(A.101a)} \\
\frac{1}{2}\sqrt{e^{2\epsilon}+2^{2/3}\sqrt[3]{e^{2\epsilon}-e^{4\epsilon}}}- & \\
\frac{1}{2}\sqrt{2e^{2\epsilon}-2^{2/3}\sqrt[3]{e^{2\epsilon}-e^{4\epsilon}}+\frac{4e^{\epsilon}-2e^{3\epsilon}}{\sqrt{e^{2\epsilon}+2^{2/3}\sqrt[3]{e^{2\epsilon}-e^{4\epsilon}}}}} & \\
-\frac{e^{\epsilon}}{2} < 0, \quad \text{if } \epsilon < \ln\sqrt{2}, & \text{(A.101b)} \\
-\frac{1}{2}\sqrt{e^{2\epsilon}+2^{2/3}\sqrt[3]{e^{2\epsilon}-e^{4\epsilon}}}+ & \\
\frac{1}{2}\sqrt{2e^{2\epsilon}-2^{2/3}\sqrt[3]{e^{2\epsilon}-e^{4\epsilon}}-\frac{4e^{\epsilon}-2e^{3\epsilon}}{\sqrt{e^{2\epsilon}+2^{2/3}\sqrt[3]{e^{2\epsilon}-e^{4\epsilon}}}}} & \\
-\frac{e^{\epsilon}}{2} > 0, \quad \text{if } \epsilon > \ln\sqrt{2}, & \text{(A.101c)} \\
-\frac{1}{2}\sqrt{e^{2\epsilon}+2^{2/3}\sqrt[3]{e^{2\epsilon}-e^{4\epsilon}}}- & \\
\frac{1}{2}\sqrt{2e^{2\epsilon}-2^{2/3}\sqrt[3]{e^{2\epsilon}-e^{4\epsilon}}-\frac{4e^{\epsilon}-2e^{3\epsilon}}{\sqrt{e^{2\epsilon}+2^{2/3}\sqrt[3]{e^{2\epsilon}-e^{4\epsilon}}}}} & \\
-\frac{e^{\epsilon}}{2} < 0, \quad \text{if } \epsilon > \ln\sqrt{2}. & \text{(A.101d)}
\end{cases}
$$

Since t is a real number and $t > 0$, we obtain Eq. (3.29).

■

A.8 Calculate the probability of a variable Y falling in the interval $[L(\epsilon, x, e^{\frac{\epsilon}{3}}), R(\epsilon, x, e^{\frac{\epsilon}{3}})]$

By replacing t in Eq. (3.25) of `PM-OPT` with $e^{\frac{\epsilon}{3}}$, we obtain the probability as follows:

$$
\begin{aligned}
&\mathbb{P}\left[L(\epsilon, x, e^{\frac{\epsilon}{3}}) \leq Y \leq R(\epsilon, x, e^{\frac{\epsilon}{3}})\right] \\
&= \int_{L(\epsilon,x,e^{\frac{\epsilon}{3}})}^{R(\epsilon,x,e^{\frac{\epsilon}{3}})} c \, \mathrm{d}Y \\
&= \int_{\frac{(e^{\epsilon}+e^{\frac{\epsilon}{3}})(xe^{\frac{\epsilon}{3}}-1)}{e^{\frac{\epsilon}{3}}(e^{\epsilon}-1)}}^{\frac{(e^{\epsilon}+e^{\frac{\epsilon}{3}})(xe^{\frac{\epsilon}{3}}+1)}{e^{\frac{\epsilon}{3}}(e^{\epsilon}-1)}} \frac{e^{\epsilon}t(e^{\epsilon}-1)}{2(t+e^{\epsilon})^2} \, \mathrm{d}Y \\
&= \frac{e^{\epsilon}}{e^{\frac{\epsilon}{3}}+e^{\epsilon}}.
\end{aligned}
$$

■

A.9 Proof of Lemma 8

The expression of the two probabilities in Eq. (A.104) can be solved from the following:

$$
\begin{cases}
\text{proper distribution so that} \\
\mathbb{P}\left[Z = \dfrac{kA}{m} \mid Y = y\right] \\
\quad + \mathbb{P}\left[Z = \dfrac{(k+1)A}{m} \mid Y = y\right] = 1, & \text{(A.102a)} \\
\\
\mathbb{E}\left[Z \mid Y = y\right] = y \text{ so that} \\
\left(\begin{array}{l} \frac{kA}{m} \times \mathbb{P}\left[Z = \frac{kA}{m} \mid Y = y\right] \\ + \frac{(k+1)A}{m} \times \mathbb{P}\left[Z = \frac{(k+1)A}{m} \mid Y = y\right] \end{array} \right) = y. & \text{(A.102b)}
\end{cases}
$$

Summarizing ① and ②, with $k := \lfloor \frac{ym}{A} \rfloor$, we have

$$
\mathbb{P}\left[Z = z \mid Y = y\right] = \begin{cases} k + 1 - \frac{ym}{A}, & \text{if } z = \frac{kA}{m}, \\ \\ \frac{ym}{A} - k, & \text{if } z = \frac{(k+1)A}{m}. \end{cases} \tag{A.103}
$$

In the perturbation step, the distribution of Y given the input x is given by

$$\mathbb{F}\left[Y = y \mid x\right] = \begin{cases} p_1, \text{ if } y \in [L(x), R(x)], \\ p_2, \text{ if } y \in [-A, L(x)) \cup (R(x), A]. \end{cases} \tag{A.104}$$

Hence, we obtain

$$\begin{aligned} &\mathbb{P}\left[Z = z \mid x\right] \\ &= \int_y \mathbb{P}\left[Z = z \mid x \text{ and } Y = y\right] \mathbb{F}\left[Y = y \mid x\right] \, \mathrm{d}y \\ &= \int_y \mathbb{P}\left[Z = z \mid Y = y\right] \mathbb{F}\left[Y = y \mid x\right] \, \mathrm{d}y. \end{aligned} \tag{A.105}$$

In addition, our mechanisms are unbiased, such that

$$\mathbb{E}[Y \mid x] = \int_y y \times \mathbb{F}\left[Y = y \mid x\right] \mathrm{d}y = x. \tag{A.106}$$

Therefore, we obtain

$$\begin{aligned} \mathbb{E}[Z \mid x] &= \sum_z z \times \mathbb{P}\left[Z = z \mid x\right] \\ &= \sum_z z \times \int_y \mathbb{P}\left[Z = z \mid Y = y\right] \mathbb{F}\left[Y = y \mid x\right] \mathrm{d}y \\ &= \int_y \left(\sum_z z \times \mathbb{P}\left[Z = z \mid Y = y\right] \right) \mathbb{F}\left[Y = y \mid x\right] \mathrm{d}y \\ &= \int_y y \times \mathbb{F}\left[Y = y \mid x\right] \mathrm{d}y = x. \end{aligned} \tag{A.107}$$

■

A.10 Proof of Lemma 9

To prove $\mathrm{Var}[Z|X = x] \geq \mathrm{Var}[Y|X = x]$, it is equivalent to prove

$$\mathbb{E}\left[Z^2 | X = x\right] - \left(\mathbb{E}\left[Z|X = x\right]\right)^2 \geq \mathbb{E}\left[Y^2 | X = x\right] - \left(\mathbb{E}\left[Y|X = x\right]\right)^2.$$

Since $\mathcal{M}_1$ and $\mathcal{M}_2$ are unbiased, we have $\mathbb{E}[Z|X=x] = \mathbb{E}[Y|X=x] = x$. Hence, it is sufficient to prove

$$\mathbb{E}\left[Z^2|X=x\right] \geq \mathbb{E}\left[Y^2|X=x\right]. \tag{A.108}$$

We derive that

$$\begin{aligned}
&\mathbb{E}\left[Z^2|X=x\right] \\
&= \sum_z z^2 \cdot \mathbb{P}\left[Z=z|X=x\right] \\
&= \sum_z z^2 \int_y \mathbb{P}\left[Z=z|Y=y\right] \cdot \mathbb{F}\left[Y=y|X=x\right] \mathrm{d}y \\
&= \int_y \sum_z z^2 \mathbb{P}\left[Z=z|Y=y\right] \cdot \mathbb{F}\left[Y=y|X=x\right] \mathrm{d}y \\
&= \int_y \mathbb{E}\left[Z^2|Y=y\right] \cdot \mathbb{F}\left[Y=y|X=x\right] \mathrm{d}y,
\end{aligned} \tag{A.109}$$

and

$$\mathbb{E}\left[Y^2|X=x\right] = \int_y y^2 \cdot \mathbb{F}\left[Y=y|X=x\right] \mathrm{d}y. \tag{A.110}$$

To prove Inequality (A.108), because of Eq. (A.109) and Eq. (A.110), it is sufficient to prove

$$\mathbb{E}\left[Z^2|Y=y\right] \geq y^2, \ \forall y \in \mathrm{Range}(Y). \tag{A.111}$$

After getting the intermediate output y from $\mathcal{M}_1$, we may discretize the intermediate output y into z_1 with probability p_1 and z_2 with probability p_2. Hereby, we have

$$p_1 + p_2 = 1.$$

Mechanism $\mathcal{M}_2$ is unbiased, so that $\mathbb{E}[Z|Y=y] = y$, and we have

$$p_1 \cdot z_1 + p_2 \cdot z_2 = y.$$

According to Cauchy–Schwarz inequality, we have

$$\begin{aligned}\mathbb{E}\left[Z^2|Y=y\right] &= p_1 \cdot z_1^2 + p_2 \cdot z_2^2 \\ &= [(\sqrt{p_1})^2 + (\sqrt{p_2})^2][(\sqrt{p_1}z_1)^2 + (\sqrt{p_2}z_2)^2] \\ &\geq (p_1 \cdot z_1 + p_2 \cdot z_2)^2 = y^2.\end{aligned}$$

Thus, we get Inequality (A.108) and Inequality (A.111).

■

A.11 Proof of Lemma 11

Given `PM-SUB`'s variance in Eq. (A.84), we have

$$\mathrm{Var}_{\mathcal{P}}[Y|x] = \frac{t+1}{e^\epsilon - 1}x^2 + \frac{(t+e^\epsilon)\big((t+1)^3 + e^\epsilon - 1\big)}{3t^2(e^\epsilon - 1)^2}. \tag{A.112}$$

Given `Three-Outputs`' variance in Eq. (A.20), we can simplify it as

$$\mathrm{Var}_{\mathcal{T}}[Y|x] = (1-a)C^2 + C^2 b|x| - x^2.$$

According to Eq. (A.21), we have $C = \frac{e^\epsilon(e^\epsilon+1)}{(e^\epsilon-1)(e^\epsilon-a)}$, so that

$$\mathrm{Var}_{\mathcal{T}}[Y|x] = \frac{(1-a)e^{2\epsilon}(e^\epsilon+1)^2}{(e^\epsilon-1)^2(e^\epsilon-a)^2} + \frac{b|x|e^{2\epsilon}(e^\epsilon+1)^2}{(e^\epsilon-1)^2(e^\epsilon-a)^2} - x^2. \tag{A.113}$$

Based on Eq. (A.112) and Eq. (A.113), we construct the variance of the hybrid mechanism as follows:

$$\begin{aligned}\mathrm{Var}_{\mathcal{H}}[Y|x] = \beta &\left(\frac{t+1}{e^\epsilon - 1}x^2 + \frac{(t+e^\epsilon)\big((t+1)^3 + e^\epsilon - 1\big)}{3t^2(e^\epsilon - 1)^2}\right) \\ &+ (1-\beta)\left(\frac{(1-a)e^{2\epsilon}(e^\epsilon+1)^2}{(e^\epsilon-1)^2(e^\epsilon-a)^2} + \frac{b|x|e^{2\epsilon}(e^\epsilon+1)^2}{(e^\epsilon-1)^2(e^\epsilon-a)^2} - x^2\right),\end{aligned}$$

where $t = e^{\frac{\epsilon}{3}}$.

From Eq. (A.21), we set $b = a \cdot \frac{e^\epsilon - 1}{e^\epsilon}$ to get the worst-case noise variance. Then, we have the variance of the hybrid mechanism as follows:

$$\begin{aligned}\mathrm{Var}_{\mathcal{H}}[Y|x] = &\left(\beta\frac{t+1}{e^\epsilon - 1} + \beta - 1\right)x^2 \\ &+ (1-\beta)\frac{ae^\epsilon(e^\epsilon+1)^2}{(e^\epsilon-1)(e^\epsilon-a)^2}|x| \\ &+ \left(\frac{(t+e^\epsilon)((t+1)^3+e^\epsilon-1)}{3t^2(e^\epsilon-1)^2}\beta\right. \\ &\left.+ (1-\beta)(1-a)\frac{e^{2\epsilon}(e^\epsilon+1)^2}{(e^\epsilon-1)^2(e^\epsilon-a)^2}\right),\end{aligned} \tag{A.114}$$

where $t = e^{\frac{\epsilon}{3}}$.

Based on Eq. (A.114), we obtain

$$\max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x] = \begin{cases} \mathrm{Var}_{\mathcal{H}}[Y|x^*], \text{if } \beta\frac{t+1}{e^\epsilon-1} + \beta - 1 < 0, 0 < x^* < 1, \\ \max\{\mathrm{Var}_{\mathcal{H}}[Y|0], \mathrm{Var}_{\mathcal{H}}[Y|1]\}, \quad \text{otherwise}, \end{cases} \tag{A.115}$$

where $x^* := \frac{(\beta-1)ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon-a)^2(\beta(e^\epsilon+t)-e^\epsilon+1)}$.

Therefore, we have the following cases to compute $\max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x]$:

(I) If $\beta\frac{t+1}{e^\epsilon-1} + \beta - 1 < 0$ and $0 < x^* < 1$, we obtain:

- $\beta < \frac{e^\epsilon-1}{e^\epsilon+t}$,
- For $x^* := \frac{(\beta-1)ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon-a)^2(\beta(e^\epsilon+t)-e^\epsilon+1)}$, if $0 < x^* < 1$, we have

$$0 < \frac{(\beta-1)ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon-a)^2(\beta(e^\epsilon+t)-e^\epsilon+1)} < 1.$$

If $\frac{(\beta-1)ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon-a)^2(\beta(e^\epsilon+t)-e^\epsilon+1)} < 1$, we have

$$\beta(2(e^\epsilon-a)^2(e^\epsilon+t) - ae^\epsilon(e^\epsilon+1)^2) < 2(e^\epsilon-a)^2(e^\epsilon-1) - ae^\epsilon(e^\epsilon+1)^2. \tag{A.116}$$

- If $2(e^\epsilon-a)^2(e^\epsilon+t) - ae^\epsilon(e^\epsilon+1)^2 > 0$, we have $\beta < \frac{2(e^\epsilon-a)^2(e^\epsilon-1)-ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon-a)^2(e^\epsilon+t)-ae^\epsilon(e^\epsilon+1)^2}$. Since $\beta < \frac{e^\epsilon-1}{e^\epsilon+t}$, by comparing $\frac{2(e^\epsilon-a)^2(e^\epsilon-1)-ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon-a)^2(e^\epsilon+t)-ae^\epsilon(e^\epsilon+1)^2}$ and $\frac{e^\epsilon-1}{e^\epsilon+t}$ as shown in Appendix A.13 to get the correct domain, we have $\frac{2(e^\epsilon-a)^2(e^\epsilon-1)-ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon-a)^2(e^\epsilon+t)-ae^\epsilon(e^\epsilon+1)^2} \leq \frac{e^\epsilon-1}{e^\epsilon+t}$. Therefore, $\beta < \frac{2(e^\epsilon-a)^2(e^\epsilon-1)-ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon-a)^2(e^\epsilon+t)-ae^\epsilon(e^\epsilon+1)^2}$.

- If $2(e^\epsilon-a)^2(e^\epsilon+t)-ae^\epsilon(e^\epsilon+1)^2=0$ and $2(e^\epsilon-a)^2(e^\epsilon-1)-ae^\epsilon(e^\epsilon+1)^2>0$, no β satisfies the condition.
- If $2(e^\epsilon-a)^2(e^\epsilon+t)-ae^\epsilon(e^\epsilon+1)^2=0$ and $2(e^\epsilon-a)^2(e^\epsilon-1)-ae^\epsilon(e^\epsilon+1)^2\leq 0$, any β satisfies the condition.
- If $2(e^\epsilon-a)^2(e^\epsilon+t)-ae^\epsilon(e^\epsilon+1)^2<0$, we have

$$\beta > \frac{2(e^\epsilon-a)^2(e^\epsilon-1)-ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon-a)^2(e^\epsilon+t)-ae^\epsilon(e^\epsilon+1)^2}.$$

By summarizing the above analysis, we have the following cases to compute $\max_{x\in[-1,1]}\mathrm{Var}_{\mathcal{H}}[Y|x]$:

1) If $2(e^\epsilon-a)^2(e^\epsilon+t)-ae^\epsilon(e^\epsilon+1)^2>0$, we have

$$\max_{x\in[-1,1]}\mathrm{Var}_{\mathcal{H}}[Y|x]=\begin{cases}\mathrm{Var}_{\mathcal{H}}[Y|x^*], \text{if } \beta<\frac{2(e^\epsilon-a)^2(e^\epsilon-1)-ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon-a)^2(e^\epsilon+t)-ae^\epsilon(e^\epsilon+1)^2},\\ \max\{\mathrm{Var}_{\mathcal{H}}[Y|0],\mathrm{Var}_{\mathcal{H}}[Y|1]\}, \quad \text{otherwise.}\end{cases}$$

2) If $2(e^\epsilon-a)^2(e^\epsilon+t)-ae^\epsilon(e^\epsilon+1)^2=0$ and $ae^\epsilon(e^\epsilon+1)^2+2(e^\epsilon-a)^2(1-e^\epsilon)>0$, we have

$$\max_{x\in[-1,1]}\mathrm{Var}_{\mathcal{H}}[Y|x]=\max\{\mathrm{Var}_{\mathcal{H}}[Y|0],\mathrm{Var}_{\mathcal{H}}[Y|1]\}.$$

3) If $2(e^\epsilon-a)^2(e^\epsilon+t)-ae^\epsilon(e^\epsilon+1)^2=0$ and $2(e^\epsilon-a)^2(e^\epsilon-1)-ae^\epsilon(e^\epsilon+1)^2\leq 0$, we have

$$\max_{x\in[-1,1]}\mathrm{Var}_{\mathcal{H}}[Y|x]=\begin{cases}\mathrm{Var}_{\mathcal{H}}[Y|x^*], \text{ if } \beta<\frac{e^\epsilon-1}{e^\epsilon+t},\\ \max\{\mathrm{Var}_{\mathcal{H}}[Y|0],\mathrm{Var}_{\mathcal{H}}[Y|1]\}, \quad \text{otherwise.}\end{cases}$$

4) If $2(e^\epsilon-a)^2(e^\epsilon+t)-ae^\epsilon(e^\epsilon+1)^2<0$, we have

$$\max_{x\in[-1,1]}\mathrm{Var}_{\mathcal{H}}[Y|x]=\begin{cases}\mathrm{Var}_{\mathcal{H}}[Y|x^*], \text{ if } \frac{2(e^\epsilon-a)^2(e^\epsilon-1)-ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon-a)^2(e^\epsilon+t)-ae^\epsilon(e^\epsilon+1)^2}<\beta<\frac{e^\epsilon-1}{e^\epsilon+t},\\ \max\{\mathrm{Var}_{\mathcal{H}}[Y|0],\mathrm{Var}_{\mathcal{H}}[Y|1]\}, \quad \text{otherwise.}\end{cases}$$

Appendix A.12 proves that

$$2(e^\epsilon-a)^2(e^\epsilon+t)-ae^\epsilon(e^\epsilon+1)^2>0$$

and Appendix A.13 proves

$$\frac{2(e^\epsilon-a)^2(e^\epsilon-1)-ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon-a)^2(e^\epsilon+t)-ae^\epsilon(e^\epsilon+1)^2}\leq\frac{e^\epsilon-1}{e^\epsilon+t}.$$

Therefore, we have

$$\max_{x\in[-1,1]}\mathrm{Var}_{\mathcal{H}}[Y|x]=\begin{cases}\mathrm{Var}_{\mathcal{H}}[Y|x^*],\text{if } 0<\beta<\frac{2(e^\epsilon-a)^2(e^\epsilon-1)-ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon-a)^2(e^\epsilon+t)-ae^\epsilon(e^\epsilon+1)^2},\\ \max\{\mathrm{Var}_{\mathcal{H}}[Y|0],\mathrm{Var}_{\mathcal{H}}[Y|1]\},\quad\text{otherwise.}\end{cases}$$

(II) Based on the above analysis, β should satisfy constraint $\frac{2(e^\epsilon-a)^2(e^\epsilon-1)-ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon-a)^2(e^\epsilon+t)-ae^\epsilon(e^\epsilon+1)^2}\leq\beta\leq 1$ in order to ensure $\max_{x\in[-1,1]}\mathrm{Var}_{\mathcal{H}}[Y=y|x]=\max\{\mathrm{Var}_{\mathcal{H}}[Y|0],\mathrm{Var}_{\mathcal{H}}[Y|1]\}$.

To get the exact value of $\max_{x\in[-1,1]}\mathrm{Var}_{\mathcal{H}}[Y|x]$, after comparing $\mathrm{Var}_{\mathcal{H}}[Y|1]$ with $\mathrm{Var}_{\mathcal{H}}[Y|0]$, values of $\mathrm{Var}_{\mathcal{H}}[Y|1]$ and $\mathrm{Var}_{\mathcal{H}}[Y|0]$ are:

- $\mathrm{Var}_{\mathcal{H}}[Y|1]=(\beta\frac{t+1}{e^\epsilon-1}+\beta-1)+(1-\beta)\frac{ae^\epsilon(e^\epsilon+1)^2}{(e^\epsilon-1)(e^\epsilon-a)^2}+\big(\frac{(t+e^\epsilon)((t+1)^3+e^\epsilon-1)}{3t^2(e^\epsilon-1)^2}\beta+(1-\beta)(1-a)\frac{e^{2\epsilon}(e^\epsilon+1)^2}{(e^\epsilon-1)^2(e^\epsilon-a)^2}\big)=\beta\big(\frac{t+1}{e^\epsilon-1}+1-\frac{ae^\epsilon(e^\epsilon+1)^2}{(e^\epsilon-1)(e^\epsilon-a)^2}+\frac{(t+e^\epsilon)((t+1)^3+e^\epsilon-1)}{3t^2(e^\epsilon-1)^2}-\frac{(1-a)e^{2\epsilon}(e^\epsilon+1)^2}{(e^\epsilon-1)^2(e^\epsilon-a)^2}\big)-1+\frac{ae^\epsilon(e^\epsilon+1)^2}{(e^\epsilon-1)(e^\epsilon-a)^2}+\frac{(1-a)e^{2\epsilon}(e^\epsilon+1)^2}{(e^\epsilon-1)^2(e^\epsilon-a)^2}$,
- $\mathrm{Var}_{\mathcal{H}}[Y|0]=\frac{(t+e^\epsilon)((t+1)^3+e^\epsilon-1)}{3t^2(e^\epsilon-1)^2}\beta+(1-\beta)(1-a)\frac{e^{2\epsilon}(e^\epsilon+1)^2}{(e^\epsilon-1)^2(e^\epsilon-a)^2}$
  $=\beta\big(\frac{(t+e^\epsilon)((t+1)^3+e^\epsilon-1)}{3t^2(e^\epsilon-1)^2}-\frac{(1-a)e^{2\epsilon}(e^\epsilon+1)^2}{(e^\epsilon-1)^2(e^\epsilon-a)^2}\big)+\frac{(1-a)e^{2\epsilon}(e^\epsilon+1)^2}{(e^\epsilon-1)^2(e^\epsilon-a)^2}$.

Since $\mathrm{Var}_{\mathcal{H}}[Y|1]$ and $\mathrm{Var}_{\mathcal{H}}[Y|0]$ are linear equations respect to β, we compare slopes of β in $\mathrm{Var}_{\mathcal{H}}[Y|1]$ and $\mathrm{Var}_{\mathcal{H}}[Y|0]$. We define the slope of β in $\mathrm{Var}_{\mathcal{H}}[Y|1]$ as

$$\begin{aligned}\mathrm{slope}_1:=&\frac{t+1}{e^\epsilon-1}+1-\frac{ae^\epsilon(e^\epsilon+1)^2}{(e^\epsilon-1)(e^\epsilon-a)^2}\\&+\frac{(t+e^\epsilon)((t+1)^3+e^\epsilon-1)}{3t^2(e^\epsilon-1)^2}-\frac{(1-a)e^{2\epsilon}(e^\epsilon+1)^2}{(e^\epsilon-1)^2(e^\epsilon-a)^2},\end{aligned}\tag{A.117}$$

and the slope of β in $\mathrm{Var}_{\mathcal{H}}[Y|0]$ as

$$\mathrm{slope}_2:=\frac{(t+e^\epsilon)((t+1)^3+e^\epsilon-1)}{3t^2(e^\epsilon-1)^2}-\frac{(1-a)e^{2\epsilon}(e^\epsilon+1)^2}{(e^\epsilon-1)^2(e^\epsilon-a)^2}.\tag{A.118}$$

Then, we represent the left boundary of β as

$$\beta_1:=\frac{2(e^\epsilon-a)^2(e^\epsilon-1)-ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon-a)^2(e^\epsilon+t)-ae^\epsilon(e^\epsilon+1)^2},\tag{A.119}$$

the right boundary of β as

$$\beta_2 := 1, \tag{A.120}$$

and the value of β at the intersection of slope_1 and slope_2 as

$$\beta_{intersection} := \frac{(c-1)(c-a)^2 - ac(c+1)^2}{(t+c)(c-a)^2 - ac(c+1)^2}. \tag{A.121}$$

Then, slope_1 and slope_2 have the following possible combinations:

1) If $\text{slope}_1 > 0$ and $\text{slope}_2 > 0$, $\beta = \beta_1$.

2) If $\text{slope}_1 < 0$ and $\text{slope}_2 < 0$, $\beta = \beta_2$.

3) If $\text{slope}_1 \cdot \text{slope}_2 < 0$ and $\beta_1 < \beta_{intersection} < \beta_2$, $\beta = \beta_{intersection}$.

4) If $\text{slope}_1 \cdot \text{slope}_2 < 0$, and $\beta_{intersection} < \beta_1$ or $\beta_{intersection} > \beta_2$, find β for $\min\big\{ \max\{\text{Var}[Y|1, \beta = \beta_1], \text{Var}[Y|0, \beta = \beta_1]\}$, $\max\{\text{Var}[Y|1, \beta = \beta_2], \text{Var}[Y|0, \beta = \beta_2]\}\big\}$.

5) If $\text{slope}_1 \cdot \text{slope}_2 = 0$,
Case 1: $\text{slope}_1 = 0$, $\text{slope}_2 \neq 0$,

   a) If $\text{slope}_1 = 0$, $\text{slope}_2 > 0$, and $\beta_{intersection} \in [\beta_1, \beta_2]$, $\beta \in [\beta_1, \beta_{intersection}]$.

   b) If $\text{slope}_1 = 0$, $\text{slope}_2 < 0$, and $\beta_{intersection} \in [\beta_1, \beta_2]$, $\beta \in [\beta_{intersection}, \beta_2]$.

   c) If $\text{slope}_1 = 0$, $\text{slope}_2 > 0$, $\beta_{intersection} < \beta_1$ or $\beta_{intersection} > \beta_2$, and $\text{Var}_{\mathcal{H}}[Y|1, \beta \in [\beta_1, \beta_2]] > \text{Var}_{\mathcal{H}}[Y|0, \beta \in [\beta_1, \beta_2]]$, $\beta \in [\beta_1, \beta_2]$.

   d) If $\text{slope}_1 = 0$, $\text{slope}_2 > 0$, $\beta_{intersection} < \beta_1$ or $\beta_{intersection} > \beta_2$, and $\text{Var}_{\mathcal{H}}[Y|0, \beta \in [\beta_1, \beta_2]] > \text{Var}_{\mathcal{H}}[Y|1, \beta \in [\beta_1, \beta_2]]$, $\beta = \beta_1$.

   e) If $\text{slope}_1 = 0$, $\text{slope}_2 < 0$, $\beta_{intersection} < \beta_1$ or $\beta_{intersection} > \beta_2$, and $\text{Var}_{\mathcal{H}}[Y|1, \beta \in [\beta_1, \beta_2]] > \text{Var}_{\mathcal{H}}[Y|0, \beta \in [\beta_1, \beta_2]]$, $\beta \in [\beta_1, \beta_2]$.

   f) If $\text{slope}_1 = 0$, $\text{slope}_2 < 0$, $\beta_{intersection} < \beta_1$ or $\beta_{intersection} > \beta_2$, and $\text{Var}_{\mathcal{H}}[Y|1, \beta \in [\beta_1, \beta_2]] < \text{Var}_{\mathcal{H}}[Y|0, \beta \in [\beta_1, \beta_2]]$, $\beta = \beta_2$.

   Case 2: $\text{slope}_2 = 0$, $\text{slope}_1 \neq 0$,

   a) If $\text{slope}_1 > 0$, $\text{slope}_2 = 0$, and $\beta_{intersection} \in [\beta_1, \beta_2]$, $\beta \in [\beta_1, \beta_{intersection}]$.

   b) If $\text{slope}_1 < 0$, $\text{slope}_2 = 0$, and $\beta_{intersection} \in [\beta_1, \beta_2]$, $\beta \in [\beta_{intersection}, \beta_2]$.

c) If $\text{slope}_2 = 0$, $\text{slope}_1 > 0$, $\beta_{intersection} < \beta_1$ or $\beta_{intersection} > \beta_2$, and $\text{Var}_{\mathcal{H}}[Y|1, \beta \in [\beta_1, \beta_2]] > \text{Var}_{\mathcal{H}}[Y|0, \beta \in [\beta_1, \beta_2]]$, $\beta = \beta_1$.

d) If $\text{slope}_2 = 0$, $\text{slope}_1 > 0$, $\beta_{intersection} < \beta_1$ or $\beta_{intersection} > \beta_2$, and $\text{Var}_{\mathcal{H}}[Y|0, \beta \in [\beta_1, \beta_2]] > \text{Var}_{\mathcal{H}}[Y|1, \beta \in [\beta_1, \beta_2]]$, $\beta \in [\beta_1, \beta_2]$.

e) If $\text{slope}_2 = 0$, $\text{slope}_1 < 0$, $\beta_{intersection} < \beta_1$ or $\beta_{intersection} > \beta_2$, and $\text{Var}_{\mathcal{H}}[Y|1, \beta \in [\beta_1, \beta_2]] > \text{Var}_{\mathcal{H}}[Y|0, \beta \in [\beta_1, \beta_2]]$, $\beta = \beta_2$.

f) If $\text{slope}_2 = 0$, $\text{slope}_1 < 0$, $\beta_{intersection} < \beta_1$ or $\beta_{intersection} > \beta_2$, and $\text{Var}_{\mathcal{H}}[Y|1, \beta \in [\beta_1, \beta_2]] < \text{Var}_{\mathcal{H}}[Y|0, \beta \in [\beta_1, \beta_2]]$, $\beta \in [\beta_1, \beta_2]$.

Case 3: $\text{slope}_1 = 0$ and $\text{slope}_2 = 0$,

a) If $\text{Var}_{\mathcal{H}}[Y|1, \beta \in [\beta_1, \beta_2]] < \text{Var}_{\mathcal{H}}[Y|0, \beta \in [\beta_1, \beta_2]]$, $\beta \in [\beta_1, \beta_2]$.

b) If $\text{Var}_{\mathcal{H}}[Y|1, \beta \in [\beta_1, \beta_2]] > \text{Var}_{\mathcal{H}}[Y|0, \beta \in [\beta_1, \beta_2]]$, $\beta \in [\beta_1, \beta_2]$.

c) If $\text{Var}_{\mathcal{H}}[Y|1, \beta \in [\beta_1, \beta_2]] = \text{Var}_{\mathcal{H}}[Y|0, \beta \in [\beta_1, \beta_2]]$, $\beta \in [\beta_1, \beta_2]$.

Proof. 1) If $\text{slope}_1 > 0$ and $\text{slope}_2 > 0$, $\text{Var}_{\mathcal{H}}[Y|1]$ and $\text{Var}_{\mathcal{H}}[Y|0]$ monotonically increase $\beta \in [\beta_1, \beta_2]$, and we obtain that $\min_\beta \max_{x\in[-1,1]} \text{Var}_{\mathcal{H}}[Y|x]$ is at $\beta = \beta_1$.

2) Similar to 1), we have $\min_\beta \max_{x\in[-1,1]} \text{Var}_{\mathcal{H}}[Y|x]$ is at $\beta = 1$.

3) If $\beta_1 < \beta_{intersection} < \beta_2$, we have following cases:

- If $\text{slope}_1 > 0$ and $\text{slope}_2 < 0$, $\text{Var}[Y|1]$ monotonically increases and $\text{Var}[Y|0]$ monotonically decreases, so when $\beta \in [\beta_1, \beta_{intersection}]$, $\max_{x\in[-1,1]} \text{Var}_{\mathcal{H}}[Y|x] = \text{Var}_{\mathcal{H}}[Y|0]$. When $\beta \in [\beta_{intersection}, \beta_2]$, $\max_{x\in[-1,1]} \text{Var}_{\mathcal{H}}[Y|x] = \text{Var}_{\mathcal{H}}[Y|1]$. Therefore, $\min_\beta \max_{x\in[-1,1]} \text{Var}_{\mathcal{H}}[Y|x] = \text{Var}_{\mathcal{H}}[Y|1] = \text{Var}_{\mathcal{H}}[Y|0]$ at $\beta = \beta_{intersection}$.

- If $\text{slope}_1 < 0$ and $\text{slope}_2 > 0$, $\text{Var}[Y|1]$ monotonically decreases and $\text{Var}[Y|0]$ monotonically increases, so when $\beta \in [\beta_1, \beta_{intersection}]$, $\max_{x\in[-1,1]} \text{Var}_{\mathcal{H}}[Y|x] = \text{Var}_{\mathcal{H}}[Y|1]$. When $\beta \in [\beta_{intersection}, \beta_2]$, $\max_{x\in[-1,1]} \text{Var}_{\mathcal{H}}[Y|x] = \text{Var}_{\mathcal{H}}[Y|0]$. Therefore, $\min_\beta \max_{x\in[-1,1]} \text{Var}_{\mathcal{H}}[Y|x] = \text{Var}_{\mathcal{H}}[Y|1] = \text{Var}_{\mathcal{H}}[Y|0]$ at $\beta = \beta_{intersection}$.

4) If $\beta_{intersection} < \beta_1$ or $\beta_{intersection} > \beta_2$, we have following cases:

- If $\text{slope}_1 > 0$, $\text{slope}_2 < 0$ and $\text{Var}_{\mathcal{H}}[Y|1] > \text{Var}_{\mathcal{H}}[Y|0]$, since $\text{Var}_{\mathcal{H}}[Y|1]$ monotonically increases in the domain, $\min_\beta \text{Var}_{\mathcal{H}}[Y|1]$ is at $\beta = \beta_1$. Thus, $\min_\beta \max_{x\in[-1,1]} \text{Var}_{\mathcal{H}}[Y|x]$ is at $\beta = \beta_1$.

- If $\text{slope}_1 > 0$, $\text{slope}_2 < 0$ and $\text{Var}_{\mathcal{H}}[Y|1] < \text{Var}_{\mathcal{H}}[Y|0]$, since $\text{Var}_{\mathcal{H}}[Y|0]$ monotonically decreases in the domain, $\min_\beta \text{Var}_{\mathcal{H}}[Y|0]$ is at $\beta = \beta_2$. Thus, $\min_\beta \max_{x\in[-1,1]} \text{Var}_{\mathcal{H}}[Y|x]$ is at $\beta = \beta_2$.

- If $\text{slope}_1 < 0$, $\text{slope}_2 > 0$ and $\text{Var}_{\mathcal{H}}[Y|1] > \text{Var}_{\mathcal{H}}[Y|0]$, $\max_{x\in[-1,1]} \text{Var}_{\mathcal{H}}[Y|x] = \text{Var}_{\mathcal{H}}[Y|1]$, since $\text{Var}_{\mathcal{H}}[Y|1]$ monotonically decreases in the domain, $\min_\beta \text{Var}_{\mathcal{H}}[Y|1]$ is at $\beta = \beta_2$. Thus, $\min_\beta \max_{x\in[-1,1]} \text{Var}_{\mathcal{H}}[Y|x]$ is at $\beta = \beta_2$.

- If $\text{slope}_1 < 0$, $\text{slope}_2 > 0$ and $\text{Var}_{\mathcal{H}}[Y|1] < \text{Var}_{\mathcal{H}}[Y|0]$, $\max_{x\in[-1,1]} \text{Var}_{\mathcal{H}}[Y|x] = \text{Var}_{\mathcal{H}}[Y|0]$, since $\text{Var}_{\mathcal{H}}[Y|0]$ monotonically increases in the domain, $\min_\beta \text{Var}_{\mathcal{H}}[Y|0]$ is at $\beta = \beta_1$. Thus, $\min_\beta \max_{x\in[-1,1]} \text{Var}_{\mathcal{H}}[Y|x]$ is at $\beta = \beta_1$.

5)

- **Case 1:**

  a) If $\text{slope}_1 = 0$, $\text{slope}_2 > 0$ and $\beta_{intersection} \in [\beta_1, \beta_2]$, we can conclude that $\beta \in [\beta_1, \beta_{intersection}]$ and $\text{Var}_{\mathcal{H}}[Y|1] > \text{Var}_{\mathcal{H}}[Y|0]$. When $\beta \in (\beta_{intersection}, \beta_2]$, $\text{Var}_{\mathcal{H}}[Y|0] > \text{Var}_{\mathcal{H}}[Y|1]$. Since $\text{Var}_{\mathcal{H}}[Y|0]$ monotonically increases if $\beta \in (\beta_{intersection}, \beta_2]$, $\min_\beta \max_{x\in[-1,1]} \text{Var}_{\mathcal{H}}[Y|x] = \text{Var}_{\mathcal{H}}[Y|1, \beta \in [\beta_1, \beta_{intersection}]]$.

  b) If $\text{slope}_1 = 0$, $\text{slope}_2 < 0$ and $\beta_{intersection} \in [\beta_1, \beta_2]$, we can conclude that $\beta \in [\beta_1, \beta_{intersection}]$ and $\text{Var}_{\mathcal{H}}[Y|0] > \text{Var}_{\mathcal{H}}[Y|1]$. When $\beta \in (\beta_{intersection}, \beta_2]$, $\text{Var}_{\mathcal{H}}[Y|1] > \text{Var}_{\mathcal{H}}[Y|0]$. Since $\text{Var}_{\mathcal{H}}[Y|1]$ does not change and $\text{Var}_{\mathcal{H}}[Y|0]$ monotonically decreases, $\min_\beta \max_{x\in[-1,1]} \text{Var}_{\mathcal{H}}[Y|x] = \text{Var}_{\mathcal{H}}[Y|0, \beta \in [\beta_{intersection}, \beta_2]]$.

  c) If $\text{slope}_1 = 0$ and $\text{slope}_2 > 0, \text{Var}_{\mathcal{H}}[Y|1, \beta \in [\beta_1, \beta_2]] > \text{Var}_{\mathcal{H}}[Y|0, \beta \in [\beta_1, \beta_2]]$ and $\beta_{intersection} > \beta_2$. Since $\text{Var}_{\mathcal{H}}[Y|1]$ does not change if $\beta \in [\beta_1, \beta_2]$, $\min_\beta \max_{x\in[-1,1]} \text{Var}_{\mathcal{H}}[Y|x] = \text{Var}_{\mathcal{H}}[Y|1, \beta \in [\beta_1, \beta_2]]$.

  d) If $\text{slope}_1 = 0$ and $\text{slope}_2 > 0, \text{Var}_{\mathcal{H}}[Y|0, \beta \in [\beta_1, \beta_2]] > \text{Var}_{\mathcal{H}}[Y|1, \beta \in [\beta_1, \beta_2]]$ and $\beta_{intersection} > \beta_2$. Since $\text{Var}_{\mathcal{H}}[Y|0]$ monotonically decreases if $\beta \in [\beta_1, \beta_2]$, $\min_\beta \max_{x\in[-1,1]} \text{Var}_{\mathcal{H}}[Y|x] = \text{Var}_{\mathcal{H}}[Y|0, \beta = \beta_2]$.

e) If $\text{slope}_1 = 0$ and $\text{slope}_2 < 0, \text{Var}_\mathcal{H}[Y|1, \beta \in [\beta_1, \beta_2]] > \text{Var}_\mathcal{H}[Y|0, \beta \in [\beta_1, \beta_2]]$ and $\beta_{intersection} < \beta_1$. Since $\text{Var}_\mathcal{H}[Y|1]$ does not change if $\beta \in [\beta_1, \beta_2]$, $\min_\beta \max_{x\in[-1,1]} \text{Var}_\mathcal{H}[Y|x] = \text{Var}_\mathcal{H}[Y|1, \beta \in [\beta_1, \beta_2]]$.

f) If $\text{slope}_1 = 0$ and $\text{slope}_2 < 0, \text{Var}_\mathcal{H}[Y|0, \beta \in [\beta_1, \beta_2]] > \text{Var}_\mathcal{H}[Y|1, \beta \in [\beta_1, \beta_2]]$ and $\beta_{intersection} > \beta_2$. Since $\text{Var}_\mathcal{H}[Y|0]$ monotonically decreases if $\beta \in [\beta_1, \beta_2]$, $\min_\beta \max_{x\in[-1,1]} \text{Var}_\mathcal{H}[Y|x] = \text{Var}_\mathcal{H}[Y|0, \beta = \beta_2]$.

- **Case 2:** The proof is similar to Case 1.
- **Case 3:** $\text{Var}_\mathcal{H}[Y|0]$ and $\text{Var}_\mathcal{H}[Y|1]$ are unchanged when $\beta \in [\beta_1, \beta_2]$. Hence, $\min_\beta \max_{x\in[-1,1]} \text{Var}_\mathcal{H}[Y|x] = \max_{x\in[-1,1]} \text{Var}_\mathcal{H}[Y|x, \beta \in [\beta_1, \beta_2]]$.

□

A.12 Proof of $2(e^\epsilon - a)^2(e^\epsilon + t) - ae^\epsilon(e^\epsilon + 1)^2 > 0$

From values of ϵ, we have the following cases:

- If $0 < \epsilon \leq \ln 5.53$, we have $2(e^\epsilon - a)^2(e^\epsilon + t) - ae^\epsilon(e^\epsilon + 1)^2 > 0$ referring to Fig. A.5.
- If $\epsilon > \ln 5.53$, we have $a = \frac{e^\epsilon}{e^\epsilon+2}$ and $t = e^{\frac{\epsilon}{3}}$, so that

$$2(e^\epsilon - a)^2(e^\epsilon + t) - ae^\epsilon(e^\epsilon + 1)^2 = \frac{e^{2\epsilon}(e^\epsilon + 1)^2(e^\epsilon - 2 + 2e^{\frac{\epsilon}{3}})}{(e^\epsilon + 2)^2}. \quad \text{(A.122)}$$

Since $\frac{e^{2\epsilon}(e^\epsilon+1)^2(-e^\epsilon+2-2e^{\frac{\epsilon}{3}})}{(e^\epsilon+2)^2} > 0$ if $\epsilon > 0$, we have $2(e^\epsilon-a)^2(e^\epsilon+t)-ae^\epsilon(e^\epsilon+1)^2 > 0$ if $\epsilon > \ln 5.53$.

Thus, we conclude that $2(e^\epsilon - a)^2(e^\epsilon + t) - ae^\epsilon(e^\epsilon + 1)^2 > 0$ if $\epsilon > 0$.

■

A.13 Proof of $\frac{2(e^\epsilon-a)^2(e^\epsilon-1)-ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon-a)^2(e^\epsilon+t)-ae^\epsilon(e^\epsilon+1)^2} \leq \frac{e^\epsilon-1}{e^\epsilon+t}$ if $t = e^{\frac{\epsilon}{3}}$

Proposition 1. $\frac{2(e^\epsilon-a)^2(e^\epsilon-1)-ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon-a)^2(e^\epsilon+t)-ae^\epsilon(e^\epsilon+1)^2} \leq \frac{e^\epsilon-1}{e^\epsilon+t}$ for $\epsilon > 0$.

Define

$$f := \frac{2(e^\epsilon - a)^2(e^\epsilon - 1) - ae^\epsilon(e^\epsilon + 1)^2}{2(e^\epsilon - a)^2(e^\epsilon + \frac{\epsilon}{3}) - ae^\epsilon(e^\epsilon + 1)^2} - \frac{e^\epsilon - 1}{e^\epsilon + e^{\frac{\epsilon}{3}}}$$
$$= \frac{ae^\epsilon(e^\epsilon + 1)^2(t + 1)}{(ae^\epsilon(e^\epsilon + 1)^2 - 2(e^\epsilon - a)^2(e^\epsilon + e^{\frac{\epsilon}{3}}))(e^\epsilon + e^{\frac{\epsilon}{3}})}, \tag{A.123}$$

and

$$h := 2(e^\epsilon - a)^2(e^\epsilon + e^{\frac{\epsilon}{3}}) - ae^\epsilon(e^\epsilon + 1)^2. \tag{A.124}$$

When $\epsilon > 0$, we have $e^\epsilon + e^{\frac{\epsilon}{3}} > 0$ and $ae^\epsilon(e^\epsilon + 1)^2(e^{\frac{\epsilon}{3}} + 1) > 0$ (the numerator of Eq. (A.123)).

- If $0 < \epsilon < \ln 2$, we have $a = 0$ and $h = -2e^{2\epsilon}(e^\epsilon + e^{\frac{\epsilon}{3}}) < 0$. Therefore, we conclude that Eq. (A.123) < 0.
- If $\ln 2 \leq \epsilon \leq \ln 5.53$, we have $a = -\frac{1}{6}(-e^{2\epsilon} - 4e^\epsilon - 5 + 2\sqrt{\Delta_0}\cos(\frac{\pi}{3} + \frac{1}{3}\arccos(-\frac{\Delta_1}{2\Delta_0^{\frac{3}{2}}})))$. Fig. A.5 shows that $h := 2(e^\epsilon - a)^2(e^\epsilon + t) - ae^\epsilon(e^\epsilon + 1)^2 < 0$, so that we obtain Eq. (A.123) < 0.
- If $\epsilon > \ln 5.53$, we have $a = \frac{e^\epsilon}{e^\epsilon + 2}$ and $h = \frac{e^{2\epsilon}(e^\epsilon + 1)^2(-2 + e^\epsilon + 2e^{\frac{\epsilon}{3}})}{(e^\epsilon + 2)^2}$. Since e^ϵ and $e^{\frac{\epsilon}{3}} > 1$, we obtain $-2 + e^\epsilon + 2e^{\frac{\epsilon}{3}} > 0$ and $h > 0$. Hence, we conclude that Eq. (A.123) < 0.

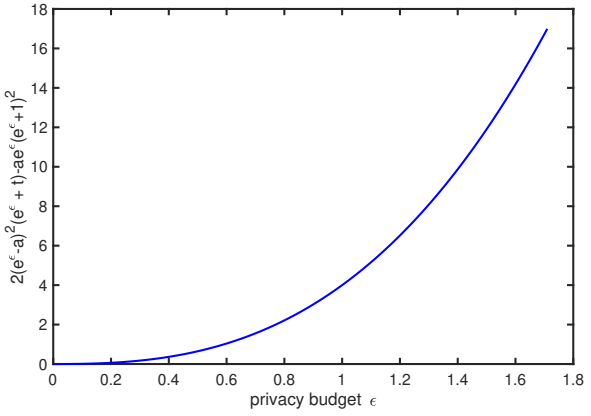


FIGURE A.5: $2(e^\epsilon - a)^2(e^\epsilon + t) - ae^\epsilon(e^\epsilon + 1)^2 > 0$ when $\epsilon \in (0, \ln 5.53]$.

Based on the above analysis, we have Eq. (A.123) < 0 when $\epsilon > 0$, meaning that $\frac{2(e^\epsilon - a)^2(e^\epsilon - 1) - ae^\epsilon(e^\epsilon + 1)^2}{2(e^\epsilon - a)^2(e^\epsilon + t) - ae^\epsilon(e^\epsilon + 1)^2} \leq \frac{e^\epsilon - 1}{e^\epsilon + t}$ when $\epsilon > 0$. ■

A.14 Proof of Lemma 10

The $\max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x]$ is minimum when

$$\beta = \begin{cases} 0, \quad \text{if} \quad 0 < \epsilon < \epsilon^*, \\ \beta_1 = \frac{2(e^\epsilon - a)^2(e^\epsilon - 1) - ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon - a)^2(e^\epsilon + t) - ae^\epsilon(e^\epsilon+1)^2}, \quad \text{if} \quad \epsilon^* \le \epsilon < \ln 2, \\ \beta_3 = \frac{-\sqrt{\frac{\omega_2}{\omega_1}} + e^\epsilon - 1}{e^\epsilon + e^{\frac{\epsilon}{3}}}, \ \text{if } \epsilon \ge \ln 2, \end{cases} \tag{A.125}$$

where

$$\epsilon^* \approx 0.610986. \tag{A.126}$$

Proof. If $x = x^* = \frac{(\beta-1)ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon-a)^2(\beta(e^\epsilon+t)-e^\epsilon+1)}$, we have the variance of Y as follows:

$$\begin{aligned} \mathrm{Var}_{\mathcal{H}}[Y|x^*] = {} & (\beta\frac{t+1}{e^\epsilon - 1} + \beta - 1)\left(\frac{(\beta-1)ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon-a)^2(\beta(e^\epsilon+t)-e^\epsilon+1)}\right)^2 \\ & + (1-\beta)\frac{ae^\epsilon(e^\epsilon+1)^2}{(e^\epsilon-1)(e^\epsilon-a)^2}\left(\frac{(\beta-1)ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon-a)^2(\beta(e^\epsilon+t)-e^\epsilon+1)}\right) \\ & + \left(\frac{(t+e^\epsilon)((t+1)^3+e^\epsilon-1)}{3t^2(e^\epsilon-1)^2}\beta\right. \\ & \left. + (1-\beta)(1-a)\frac{e^{2\epsilon}(e^\epsilon+1)^2}{(e^\epsilon-1)^2(e^\epsilon-a)^2}\right). \end{aligned} \tag{A.127}$$

Let $\gamma := \beta(e^\epsilon + t) - e^\epsilon + 1$ and $c := e^\epsilon$, and then we transform $\mathrm{Var}_{\mathcal{H}}[Y|x^*]$ in Eq. (A.127) to the following:

$$\begin{aligned}
&\gamma\Bigg(\frac{a^2c^2(c+1)^4}{4(c+t)^2(c-a)^4(c-1)}-\frac{a^2c^2(c+1)^4}{2(c+t)^2(c-1)(c-a)^4}\\
&\quad+\frac{(t+1)^3+c-1}{3t^2(c-1)^2}-\frac{(1-a)c^2(c+1)^2}{(c+t)(c-1)^2(c-a)^2}\Bigg)\\
&\quad+\frac{1}{\gamma}\Bigg(\frac{(1+t)^2a^2c^2(c+1)^4}{4(c+t)^2(c-a)^4(c-1)}-\frac{(1+t)^2a^2c^2(c+1)^4}{2(c+t)^2(c-1)(c-a)^4}\Bigg)\\
&\quad-\frac{(1+t)a^2c^2(c+1)^4}{2(c+t)^2(c-a)^4(c-1)}+\frac{(1+t)a^2c^2(c+1)^4}{(c+t)^2(c-1)(c-a)^4}\\
&\quad+\frac{(t+1)^3+c-1}{3t^2(c-1)}+\frac{(1+t)(1-a)c^2(c+1)^2}{(c+t)(c-1)^2(c-a)^2}.
\end{aligned}\tag{A.128}$$

Set coefficient of γ as ω_1 :

$$\begin{aligned}
\omega_1 :=\ &\frac{a^2c^2(c+1)^4}{4(c+t)^2(c-a)^4(c-1)}-\frac{a^2c^2(c+1)^4}{2(c+t)^2(c-1)(c-a)^4}\\
&+\frac{(t+1)^3+c-1}{3t^2(c-1)^2}-\frac{(1-a)c^2(c+1)^2}{(c+t)(c-1)^2(c-a)^2}.
\end{aligned}\tag{A.129}$$

Set coefficient of $\frac{1}{\gamma}$ as ω_2 :

$$\omega_2 := -\frac{(1+t)^2a^2c^2(c+1)^4}{4(c+t)^2(c-a)^4(c-1)}.\tag{A.130}$$

Set the rest of (A.128) as:

$$\begin{aligned}
\omega_3 :=\ &-\frac{(1+t)a^2c^2(c+1)^4}{2(c+t)^2(c-a)^4(c-1)}+\frac{(1+t)a^2c^2(c+1)^4}{(c+t)^2(c-1)(c-a)^4}\\
&+\frac{(t+1)^3+c-1}{3t^2(c-1)}+\frac{(1+t)(1-a)c^2(c+1)^2}{(c+t)(c-1)^2(c-a)^2}.
\end{aligned}\tag{A.131}$$

Since γ monotonically increases with β in the domain $\beta \in (0, \beta_1)$, the minimum γ is $-e^\epsilon + 1$ at $\beta = 0$, and maximum γ is at $\beta = \beta_1$.

- If $0 < \epsilon < \ln 2$, $a = 0$, and we have

$$\begin{aligned}
\omega_1 &= \frac{(t+1)^3+c-1}{3t^2(c-1)^2}-\frac{(c+1)^2}{(c+t)(c-1)^2},\\
\omega_2 &= 0,\\
&\text{and } \mathrm{Var}_{\mathcal{H}}[Y|x^*] = \omega_1\gamma+\omega_3 \text{ is a linear function.}
\end{aligned}$$

-If $\beta \in (0, \beta_1)$, Appendix A.16 proves:

a) $\omega_1 > 0$, if $0 < \epsilon < 0.610986$.

b) $\omega_1 = 0$, if $\epsilon = 0.610986$.

c) $\omega_1 < 0$, if $0.610986 < \epsilon < \beta_1$.

Therefore, $\min_\beta \max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x]$ is at:

a) $\beta = 0$, if $0 < \epsilon < 0.610986$.

b) $\beta = \beta_1$, if $0.610986 \leq \epsilon < \ln 2$.

-If $\beta \in [\beta_1, \beta_2]$, $\mathrm{slope}_1 = \frac{t+1}{e^\epsilon - 1} + 1 + \frac{(t+e^\epsilon)((t+1)^3+e^\epsilon-1)}{3t^2(e^\epsilon-1)^2} - \frac{(e^\epsilon+1)^2}{(e^\epsilon-1)^2}$, and $\mathrm{slope}_2 = \frac{(t+e^\epsilon)((t+1)^3+e^\epsilon-1)}{3t^2(e^\epsilon-1)^2} - \frac{(e^\epsilon+1)^2}{(e^\epsilon-1)^2}$.
Fig. A.6 proves that $\mathrm{slope}_1 > 0$, when $\epsilon \in [0, \ln 2]$.

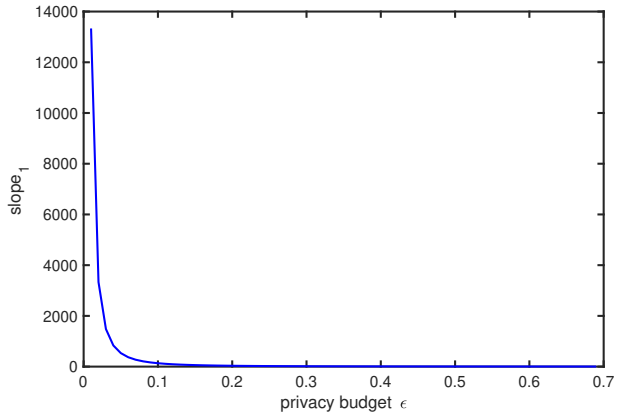


FIGURE A.6: slope_1 when $\epsilon \in [0, \ln 2]$.

When $a = 0$ and β_1 $(A.119) = \beta_{intersection}$ $(A.121) = \frac{e^\epsilon - 1}{t+e^\epsilon}$, so the intersection of $\mathrm{Var}_{\mathcal{H}}[Y|0]$ and $\mathrm{Var}_{\mathcal{H}}[Y|1]$ is at β_1.

When $\mathrm{slope}_2 = 0$, we have root at $\epsilon \approx 0.610986$.

From Fig. A.7, we have

(1) If $0 < \epsilon < 0.610986$, $\mathrm{slope}_2 > 0$.

(2) If $\epsilon = 0.610986$, $\mathrm{slope}_2 = 0$.

(3) If $\epsilon > 0.610986$, $\mathrm{slope}_2 < 0$.

Based on the previous analysis, we have

(1) If $0 < \epsilon < 0.610986$, $\min_\beta \max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x] = \max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x, \beta = \beta_1]$.

(2) If $\epsilon = 0.610986$, $\min_\beta \max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x] = \mathrm{Var}_{\mathcal{H}}[Y|1, \beta = \beta_1]$.

(3) If $\epsilon > 0.610986$, $\min_\beta \max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x] = \mathrm{Var}_{\mathcal{H}}[Y|1, \beta = \beta_1]$.

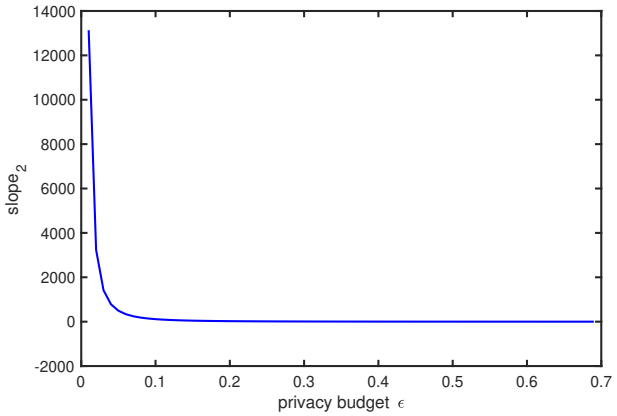


FIGURE A.7: slope_2 when $\epsilon \in [0, \ln 2]$.

Therefore, $\min_\beta \max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x]$ is at $\beta = \beta_1$ if $\beta \in [\beta_1, \beta_2]$. By summarizing the above analysis, we can conclude that $\min_\beta \max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x] =$

- $\mathrm{Var}_{\mathcal{H}}[Y|x^*, \beta = 0]$, if $0 < \epsilon < 0.610986$.
- $\max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x, \beta = \beta_1]$, if $0.610986 \leq \epsilon < \ln 2$.

- If $\ln 2 \leq \epsilon \leq \ln 5.53$,
  (1) when $\beta \in (0, \beta_1)$, Appendix A.16 proves $\omega_1 < 0$ and $\omega_2 < 0$. Appendix A.15 proves that when $\gamma = -\sqrt{\frac{\omega_2}{\omega_1}}$, we have

$$\min_\beta \max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x] = \max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x, \beta = \beta_3].$$

  Since $\gamma := \beta(c+t) - c + 1$, we have $\beta_3 := \frac{-\sqrt{\frac{\omega_2}{\omega_1}} + c - 1}{c+t}$. $\min_\beta \max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x] = \max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x^*, \beta = \beta_3]$.
  (2) When $\beta \in [\beta_1, \beta_2]$, Fig. A.8 shows that $\text{slope}_1 > 0$ if $\epsilon \in [\ln 2, \ln 5.53]$. Fig. A.9 shows values of slope_2, and we have following cases:

  - If $0 < \epsilon < 1.4338$ and $\text{slope}_2 < 0$, $\beta_{intersection} < \beta_1$ as shown in Fig. A.10, and

$$\min_\beta \max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x] = \max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x, \beta = \beta_1].$$

– If $\epsilon \approx 1.4338$ and $\text{slope}_2 = 0$, $\beta_{intersection} < \beta_1$ as shown in Fig. A.10, and

$$\min_{\beta} \max_{x\in[-1,1]} \text{Var}_{\mathcal{H}}[Y|x] = \max_{x\in[-1,1]} \text{Var}_{\mathcal{H}}[Y|x, \beta = \beta_1].$$

– If $1.4338 < \epsilon \leq \ln 5.53$ and $\text{slope}_2 > 0$, $\min_\beta \max_{x\in[-1,1]} \text{Var}_{\mathcal{H}}[Y|x] = \max_{x\in[-1,1]} \text{Var}_{\mathcal{H}}[Y|x, \beta = \beta_1]$.

Since $\text{Var}_{\mathcal{H}}[Y|x, \beta = \beta_3] < \text{Var}_{\mathcal{H}}[Y|x, \beta = \beta_1]$, $\min_\beta \max_{x\in[-1,1]} \text{Var}_{\mathcal{H}}[Y|x]$ is at $\beta = \beta_3$.

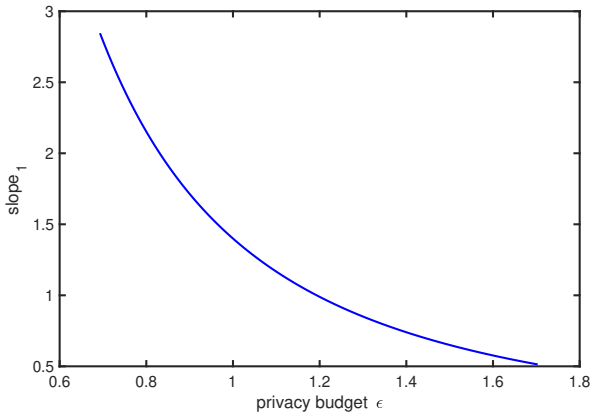


FIGURE A.8: slope_1 when $\epsilon \in [\ln 2, \ln 5.53]$.

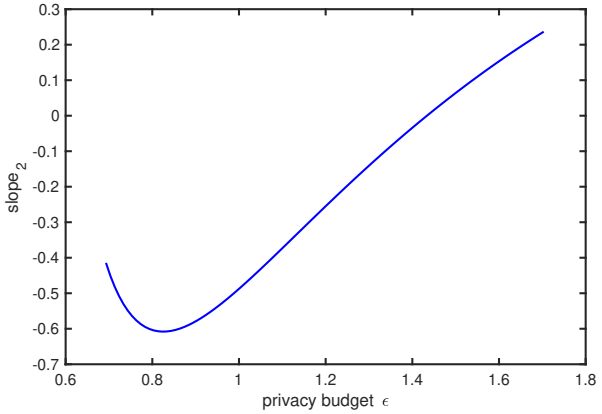


FIGURE A.9: slope_2 when $\epsilon \in [\ln 2, \ln 5.53]$.

- If $\epsilon > \ln 5.53$, we have following cases:

  – If $\beta \in (0, \beta_1)$, Appendix A.16 proves $\omega_1 < 0$ and $\omega_2 < 0$, and Appendix A.15 proves when $\gamma = -\sqrt{\frac{\omega_2}{\omega_1}}$, $\beta_4 := \frac{-\sqrt{\frac{\omega_2}{\omega_1}}+c-1}{c+t}$. Therefore, we

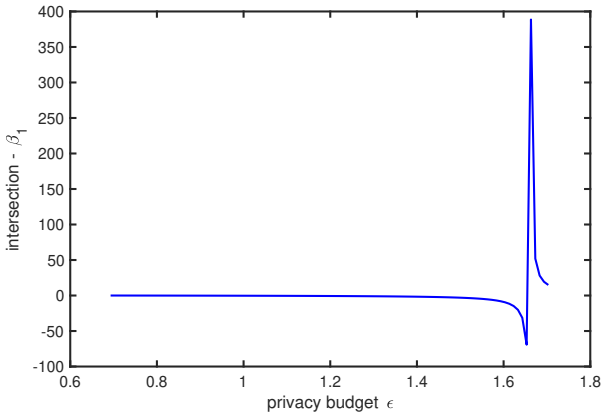


FIGURE A.10: $\beta_{intersection} - \beta_1$ when $\epsilon \in [\ln 2, \ln 5.53]$.

obtain

$$\min_{\beta} \max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x] = \max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x, \beta = \beta_4].$$

– If $\beta \in [\beta_1, \beta_2]$, Appendix A.17 proves $\text{slope}_1 > 0$ and Appendix A.18 proves $\text{slope}_2 > 0$. Therefore, we obtain

$$\min_{\beta} \max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x] = \max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x, \beta = \beta_1].$$

■

A.15 Proof of the monotonicity of $\mathbf{Var}_{\mathcal{H}}[Y|x^*]$

Substituting ω_1 (Eq. A.129), ω_2 (Eq. A.130) and ω_3 (Eq. A.131) into $\mathrm{Var}_{\mathcal{H}}[Y|x^*]$ (Eq. A.128) yields

$$\mathrm{Var}_{\mathcal{H}}[Y|x^*] = \omega_1\gamma + \frac{\omega_2}{\gamma} + \omega_3. \tag{A.132}$$

The first-order derivative of (A.132) is

$$\mathrm{Var}_{\mathcal{H}}[Y|x^*]' = \omega_1 - \frac{\omega_2}{\gamma^2}. \tag{A.133}$$

If $\omega_1 - \frac{\omega_2}{\gamma^2} = 0$, we get two roots:

$$\gamma_1 = -\sqrt{\frac{\omega_2}{\omega_1}},\ \gamma_2 = \sqrt{\frac{\omega_2}{\omega_1}}.$$

Hereby, we have the following cases:

1) If $\gamma \in (-\infty, -\sqrt{\frac{\omega_2}{\omega_1}}]$, $\mathrm{Var}_{\mathcal{H}}[Y|x^*]' < 0$, so that $\mathrm{Var}_{\mathcal{H}}[Y|x^*]$ monotonically decreases.

2) If $\gamma \in (-\sqrt{\frac{\omega_2}{\omega_1}}, 0)$, $\mathrm{Var}_{\mathcal{H}}[Y|x^*]' > 0$, so that $\mathrm{Var}_{\mathcal{H}}[Y|x^*]$ monotonically increases.

3) If $\gamma \in (0, \sqrt{\frac{\omega_2}{\omega_1}}]$, $\mathrm{Var}_{\mathcal{H}}[Y|x^*]' > 0$, so that $\mathrm{Var}_{\mathcal{H}}[Y|x^*]$ monotonically increases.

4) If $\gamma \in (\sqrt{\frac{\omega_2}{\omega_1}}, +\infty)$, $\mathrm{Var}_{\mathcal{H}}[Y|x^*]' < 0$, so that $\mathrm{Var}_{\mathcal{H}}[Y|x^*]$ monotonically decreases.

If $\gamma > 0$, according to cases 3) and 4), the minimum $\mathrm{Var}_{\mathcal{H}}[Y|x^*] = -\infty < 0$ because $\omega_1, \omega_2 < 0$ when $\epsilon \geq \ln 2$, which violates the definition of the variance. The value of variance is larger than 0 according to its formal definition. The minimum value of $\mathrm{Var}_{\mathcal{H}}[Y|x^*]$ can be achieved only when γ is close to 0 or $+\infty$. This implies that $\mathrm{Var}_{\mathcal{H}}[Y|x^*]$ is < 0. Thus, $\gamma = \gamma_1 = -\sqrt{\frac{\omega_2}{\omega_1}} < 0$ is eligible.

■

A.16 The sign of ω_1 to ϵ

If $\omega_1 = 0$, we obtain $\epsilon \approx 0.610986$.

The first-order derivative of ω_1 is

$$\begin{aligned}\omega_1' = &-(25e^{\epsilon} - 27e^{2\epsilon} - 9e^{3\epsilon} - 12e^{\frac{\epsilon}{3}} + 19e^{2\epsilon/3} - e^{4\epsilon/3} \\ &+ 41e^{5\epsilon/3} + 7e^{7\epsilon/3} + 5)/(9e^{2\epsilon/3}(e^{2\epsilon/3} + 1)^2(e^{\epsilon} - 1)^3).\end{aligned} \tag{A.134}$$

- If $0 < \epsilon < \ln 2$, Fig. A.11 shows that $\omega_1' < 0$ and ω_1 monotonically decreases if $\epsilon \in (0, \ln 2)$. Therefore, we have following cases:

- $\omega_1 > 0$, if $0 < \epsilon < 0.610986$.
- $\omega_1 = 0$, if $\epsilon = 0.610986$.
- $\omega_1 < 0$, if $0.610986 < \epsilon < \ln 2$.

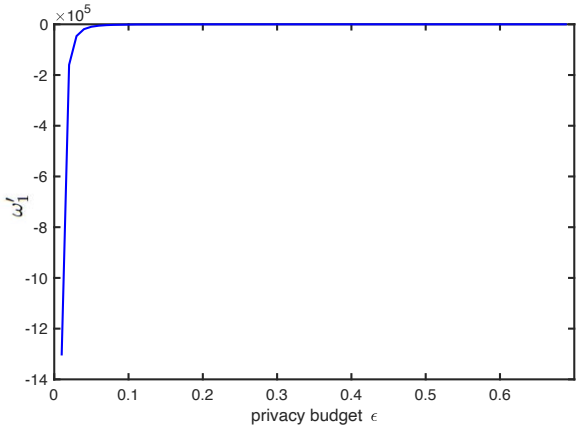


FIGURE A.11: The first-order derivative of ω_1 is less than 0 when $0 < \epsilon \leq \ln 2$.

- If $\ln 2 \leq \epsilon \leq \ln 5.53$, Fig. A.12 shows $\omega_1 < 0$.

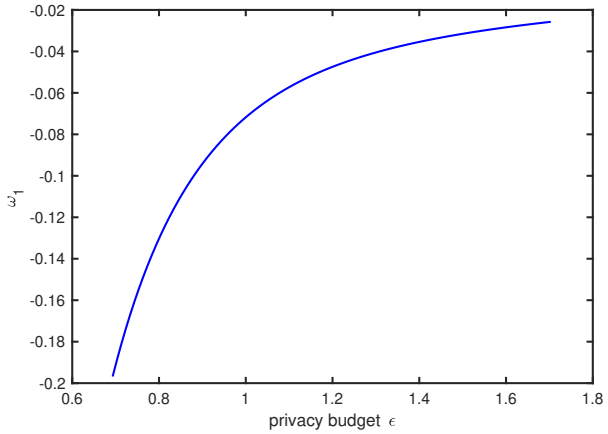


FIGURE A.12: ω_1 is less than 0 if $\ln 2 < \epsilon \leq \ln 5.53$.

- If $\epsilon > \ln 5.53$, we obtain

$$\omega_1 = -(16e^{\epsilon} + 21e^{2\epsilon} + 3e^{3\epsilon} + 36e^{\frac{\epsilon}{3}} - 12e^{\frac{2\epsilon}{3}} - 28e^{\frac{5\epsilon}{3}} - 8e^{\frac{7\epsilon}{3}} - 12)/(12e^{\frac{2\epsilon}{3}}(e^{\epsilon} - e^{\frac{2\epsilon}{3}} + e^{\frac{5\epsilon}{3}} - 1)^2). \tag{A.135}$$

When $\omega_1 = 0$, we have three roots:

$$r_1 \approx -16.9563, r_2 \approx -1.2284, r_3 \approx 0.0463914.$$

If the denominator is

$$12e^{\frac{2\epsilon}{3}}(e^{\epsilon} - e^{\frac{2\epsilon}{3}} + e^{\frac{5\epsilon}{3}} - 1)^2 > 0$$

and

$$\lim_{\epsilon\to\infty} -(16e^{\epsilon} + 21e^{2\epsilon} + 3e^{3\epsilon} + 36e^{\frac{\epsilon}{3}} - 12e^{\frac{2\epsilon}{3}} - 28e^{\frac{5\epsilon}{3}} - 8e^{\frac{7\epsilon}{3}} - 12) = -\infty,$$

r_3 is the largest real value root and the sign of

$$-(16e^{\epsilon} + 21e^{2\epsilon} + 3e^{3\epsilon} + 36e^{\frac{\epsilon}{3}} - 12e^{\frac{2\epsilon}{3}} - 28e^{\frac{5\epsilon}{3}} - 8e^{\frac{7\epsilon}{3}} - 12)$$

does not change, so that $\omega_1 < 0$ when $\epsilon > \ln 5.53$.

■

A.17 The sign of slope$_1$ when $\epsilon > \ln 5.53$

When $\epsilon > \ln 5.53$, $a = \frac{e^{\epsilon}}{e^{\epsilon}+2}$ and

$$\begin{aligned}\text{slope}_1 = \frac{t+1}{e^{\epsilon}-1} + 1 - \frac{ae^{\epsilon}(e^{\epsilon}+1)^2}{(e^{\epsilon}-1)(e^{\epsilon}-a)^2} \\ + \frac{(t+e^{\epsilon})((t+1)^3+e^{\epsilon}-1)}{3t^2(e^{\epsilon}-1)^2} - \frac{(1-a)e^{2\epsilon}(e^{\epsilon}+1)^2}{(e^{\epsilon}-1)^2(e^{\epsilon}-a)^2}.\end{aligned} \tag{A.136}$$

The first-order derivative of slope$_1$ is

$$\text{slope}_1' = -\frac{e^{\frac{2\epsilon}{3}}\ (10\,e^{\epsilon}+20) + 20\,e^{\epsilon} + (-27\,e^{\epsilon}-45)\ e^{\frac{\epsilon}{3}} + 10}{9e^{\frac{\epsilon}{3}}\ (e^{\epsilon}-1)^3}.$$

The denominator of slope$_1'$ is > 0. Thus, we obtain

$$\lim_{\epsilon\to\infty} -e^{\frac{2\epsilon}{3}}\ (10\,e^{\epsilon}+20) + 20\,e^{\epsilon} + (-27\,e^{\epsilon}-45)\ e^{\frac{\epsilon}{3}} + 10 = -\infty.$$

If slope$_1' = 0$, we have two roots:

$$e_1^{\epsilon} \approx 0.0169067, e_2^{\epsilon} \approx 4.22192.$$

Since $e_2^\epsilon \approx 4.22192$ is the largest real value root, the sign of slope_1' does not change when $e^\epsilon > 4.22192$. Therefore, when $\epsilon > \ln 5.53$ and $\text{slope}_1' < 0$, slope_1 monotonically decreases.

After simplifying slope_1, we get

$$\text{slope}_1 = \frac{-(9e^\epsilon - 5e^{2\epsilon/3} - 5e^{4\epsilon/3} + 3)}{3(e^\epsilon - 1)^2}.$$

Then, we obtain $\lim_{\epsilon\to\infty} \text{slope}_1 = 0$. Thus, we have $\text{slope}_1 > 0$ if $\epsilon > \ln 5.53$.

■

A.18 The sign of slope_2 when $\epsilon > \ln 5.53$

When $\epsilon > \ln 5.53$, $a = \frac{e^\epsilon}{e^\epsilon+2}$ and

$$\text{slope}_2 = \frac{(t+e^\epsilon)((t+1)^3 + e^\epsilon - 1)}{3t^2(e^\epsilon-1)^2} - \frac{(1-a)e^{2\epsilon}(e^\epsilon+1)^2}{(e^\epsilon-1)^2(e^\epsilon-a)^2}. \tag{A.137}$$

The first-order derivative of slope_2 is

$$\text{slope}_2' = \frac{-4\,e^{2\epsilon} + e^{\frac{2\epsilon}{3}}\,(9\,e^\epsilon + 63) - 23\,e^\epsilon + (-20\,e^\epsilon - 10)\,e^{\frac{\epsilon}{3}} - 3}{9e^{\frac{2\epsilon}{3}}\,(e^\epsilon - 1)^3}. \tag{A.138}$$

The denominator of Eq. (*A*.138) is > 0. Besides, from nominator of Eq. (*A*.138), we obtain

$$\lim_{\epsilon\to\infty}(-4\,e^{2\epsilon} + e^{\frac{2\epsilon}{3}}\,(9\,e^\epsilon + 63) - 23\,e^\epsilon + (-20\,e^\epsilon - 10)\,e^{\frac{\epsilon}{3}} - 3) = -\infty.$$

If Eq. (*A*.138) $= 0$, we have one root:

$$\epsilon \approx 0.709472.$$

The sign of slope_2' does not change if $\epsilon > 0.709472$. Therefore, $\text{slope}_2' < 0$ if $\epsilon > \ln 5.53$. Simplify

$$\text{slope}_2 = \frac{3e^{\frac{\epsilon}{3}} - 3e^\epsilon + 5e^{\frac{2\epsilon}{3}} + 2e^{\frac{4\epsilon}{3}} - 9}{3(e^\epsilon - 1)^2},$$

and then we obtain

$$\lim_{\epsilon \to \infty} \text{slope}_2 = 0.$$

Thus, we have $\text{slope}_2 > 0$ if $\epsilon > \ln 5.53$.

■

A.19 Proof of Lemma 13

For any $i \in [1, n]$, the random variable $Y[t_j] - x[t_j]$ has zero mean based on Lemma 12. In both `PM-SUB` and $\texttt{HM}_{\text{PM-SUB,Three-Outputs}}$, $|Y[t_j] - x[t_j]| \leq \frac{d}{k} \cdot \frac{(e^{\frac{\epsilon}{k}} + e^{\frac{\epsilon}{3k}})(e^{\frac{\epsilon}{3k}} + 1)}{e^{\frac{\epsilon}{3k}}(e^{\frac{\epsilon}{k}} - 1)}$.

According to Bernstein's inequality, we have

$$\begin{aligned}
&\mathbb{P}\left[|Z[t_j] - X[t_j]| \geq \lambda\right] \\
&= \mathbb{P}\left[|\sum_{i=1}^{n} \{Y[t_j] - x[t_j]\}| \geq n\lambda\right] \\
&\leq 2 \cdot \exp\left(\frac{-(n\lambda)^2}{2\sum_{i=1}^{n} \text{Var}[Y[t_j]] + \frac{2}{3} \cdot n\lambda \cdot \frac{d}{k} \cdot \frac{(e^{\frac{\epsilon}{k}} + e^{\frac{\epsilon}{3k}})(e^{\frac{\epsilon}{3k}} + 1)}{e^{\frac{\epsilon}{3k}}(e^{\frac{\epsilon}{k}} - 1)}}\right).
\end{aligned}$$

In Algorithm 6, $Y[t_j]$ equals $\frac{d}{k} y_j$ with probability $\frac{k}{d}$ and 0 with probability $1 - \frac{k}{d}$. Moreover, we obtain $\mathbb{E}[Y[t_j]] = x[t_j]$ from Lemma 12, and then we get

$$\begin{aligned}
\text{Var}[Y[t_j]] &= \mathbb{E}[(Y[t_j])^2] - \mathbb{E}[Y[t_j]]^2 \\
&= \frac{k}{d} \cdot \mathbb{E}[(\frac{d}{k} y_j)^2] - (x[t_j])^2 \\
&= \frac{d}{k} \mathbb{E}[(y_j)^2] - (x[t_j])^2. \qquad \text{(A.139)}
\end{aligned}$$

In Algorithm 6, if Line 5 uses `PM-SUB`, we obtain the variance in Eq. (A.84) with the privacy budget $\frac{\epsilon}{k}$ to compute $\mathbb{E}[(y_j)^2]$, and the asymptotic expression involving

ϵ is in the sense of $\epsilon \to 0$.

$$\begin{aligned}\mathbb{E}[(y_j)^2] &= \mathrm{Var}[y_j] + (\mathbb{E}[y_j])^2 \\ &= \frac{t+1}{e^{\frac{\epsilon}{k}}-1}(x[t_j])^2 + \frac{(t+e^{\frac{\epsilon}{k}})\big((t+1)^3+e^{\frac{\epsilon}{k}}-1\big)}{3t^2(e^{\frac{\epsilon}{k}}-1)^2} + (x[t_j])^2 \\ &= O\left(\frac{k^2}{\epsilon^2}\right). \end{aligned} \tag{A.140}$$

In Algorithm 6, if Line 5 uses `Three-Outputs`, we use the variance in Eq. (A.113) to compute $\mathbb{E}[(y_j)^2]$ below, so that the asymptotic expression involving ϵ is in the sense of $\epsilon \to 0$.

$$\begin{aligned}\mathbb{E}[(y_j)^2] &= \mathrm{Var}[y_j] + (\mathbb{E}[y_j])^2 \\ &= \frac{(1-a)e^{\frac{2\epsilon}{k}}(e^{\frac{\epsilon}{k}}+1)^2}{(e^{\frac{\epsilon}{k}}-1)^2(e^{\frac{\epsilon}{k}}-a)^2} + \frac{b|x[t_j]|e^{\frac{2\epsilon}{k}}(e^{\frac{\epsilon}{k}}+1)^2}{(e^{\frac{\epsilon}{k}}-1)^2(e^{\frac{\epsilon}{k}}-a)^2} - (x[t_j])^2 + (x[t_j])^2 \\ &= \frac{(1-a)e^{\frac{2\epsilon}{k}}(e^{\frac{\epsilon}{k}}+1)^2}{(e^{\frac{\epsilon}{k}}-1)^2(e^{\frac{\epsilon}{k}}-a)^2} + \frac{b|x[t_j]|e^{\frac{2\epsilon}{k}}(e^{\frac{\epsilon}{k}}+1)^2}{(e^{\frac{\epsilon}{k}}-1)^2(e^{\frac{\epsilon}{k}}-a)^2} \\ &= O\left(\frac{k^2}{\epsilon^2}\right). \end{aligned}$$

In Algorithm 6, if Line 5 uses `HM-TP`, we have

$$\begin{aligned}\mathbb{E}[(y_j)^2] &= \mathrm{Var}[y_j] + (\mathbb{E}[y_j])^2 \\ &= \begin{cases} \frac{(e^{\frac{\epsilon}{k}}+1)^2}{(e^{\frac{\epsilon}{k}}-1)^2} + (x[t_j])^2, & \text{If } 0 < \epsilon < \epsilon^*, \\ \mathrm{Var}_{\mathcal{H}}[Y|1,\beta_1,\frac{\epsilon}{k}] + (x[t_j])^2, & \text{If } \epsilon^* \le \epsilon < \ln 2, \\ \mathrm{Var}_{\mathcal{H}}[Y|1,\beta_3,\frac{\epsilon}{k}] + (x[t_j])^2, & \text{If } \epsilon \ge \ln 2 \end{cases} \\ &= O\left(\frac{k^2}{\epsilon^2}\right), \end{aligned} \tag{A.141}$$

where ϵ^* is defined in the Eq. (A.126).

Then, we obtain

$$\begin{aligned}\mathrm{Var}[Y[t_j]] = \frac{d}{k}\cdot\Bigg(&\frac{t+1}{e^{\frac{\epsilon}{k}}-1}(x[t_j])^2 \\ &+ \frac{(t+e^{\frac{\epsilon}{k}})\big((t+1)^3+e^{\frac{\epsilon}{k}}-1\big)}{3t^2(e^{\frac{\epsilon}{k}}-1)^2} + (x[t_j])^2\Bigg) \\ &- (x[t_j])^2. \end{aligned} \tag{A.142}$$

Substituting Eq. (A.140) into Eq. (A.139) yields

$$\mathrm{Var}[Y[t_j]] = \frac{d}{k}\cdot O\left(\frac{k^2}{\epsilon^2}\right) - (x[t_j])^2 = O\left(\frac{dk}{\epsilon^2}\right). \tag{A.143}$$

Therefore, we obtain

$$\mathbb{P}\left[|Z[t_j] - X[t_j]| \geq \lambda\right] \leq 2\cdot\exp\left(-\frac{n\lambda^2}{O(dk/\epsilon^2)+\lambda\cdot O(d/\epsilon)}\right).$$

By leveraging the union bound, there exists $\lambda = O\left(\frac{\sqrt{d\ln(d/\beta))}}{\epsilon\sqrt{n}}\right)$. Therefore, $\max_{j\in[1,d]}|Z[t_j] - X[t_j]| = \lambda = O\left(\frac{\sqrt{d\ln(d/\beta))}}{\epsilon\sqrt{n}}\right)$.

■

A.20 Calculate k for `PM-SUB` and `Three-Outputs`

We calculate the optimal k for `PM-SUB` and `Three-Outputs` when numeric data are multidimensional.

(I) For `PM-SUB`, we obtain

$$\max_{x[t_j]\in[-1,1]}\mathrm{Var}[Y[t_j]] = \frac{d}{k}\left(\frac{t+1}{e^{\frac{\epsilon}{k}}-1} + \frac{(t+e^{\frac{\epsilon}{k}})\big((t+1)^3+e^{\frac{\epsilon}{k}}-1\big)}{3t^2(e^{\frac{\epsilon}{k}}-1)^2} + 1\right) - 1, \tag{A.144}$$

when $x[t_j] = 1$.

Then, we substitute $t = e^{\frac{\epsilon}{3k}}$ into Eq. (A.144) and obtain

$$\begin{aligned}&\max_{x[t_j]\in[-1,1]} \mathrm{Var}[Y[t_j]] \\ &= \frac{d}{k}\left(\frac{e^{\frac{\epsilon}{3k}}+1}{e^{\frac{\epsilon}{k}}-1} + \frac{(e^{\frac{\epsilon}{3k}}+e^{\frac{\epsilon}{k}})\left((e^{\frac{\epsilon}{3k}}+1)^3+e^{\frac{\epsilon}{k}}-1\right)}{3(e^{\frac{\epsilon}{3k}})^2(e^{\frac{\epsilon}{k}}-1)^2}+1\right) \\ &- 1.\end{aligned} \tag{A.145}$$

Let $s = \frac{\epsilon}{k}$, and we achieve

$$\begin{aligned}&\max_{x[t_j]\in[-1,1]} \mathrm{Var}[Y[t_j]] \\ &= \frac{d}{\epsilon}\cdot s\left(\frac{e^{\frac{s}{3}}+1}{e^{s}-1} + \frac{(e^{\frac{s}{3}}+e^{s})\left((e^{\frac{s}{3}}+1)^3+e^{s}-1\right)}{3(e^{\frac{s}{3}})^2(e^{s}-1)^2}+1\right) \\ &- 1.\end{aligned} \tag{A.146}$$

Let

$$f(s) = s\cdot\left(\frac{e^{\frac{s}{3}}+1}{e^{s}-1} + \frac{(e^{\frac{s}{3}}+e^{s})\left((e^{\frac{s}{3}}+1)^3+e^{s}-1\right)}{3(e^{\frac{s}{3}})^2(e^{s}-1)^2}+1\right),$$

and then we get

$$\max_{x[t_j]\in[-1,1]} \mathrm{Var}[Y[t_j]] = \frac{d}{\epsilon}\cdot f(s) - 1. \tag{A.147}$$

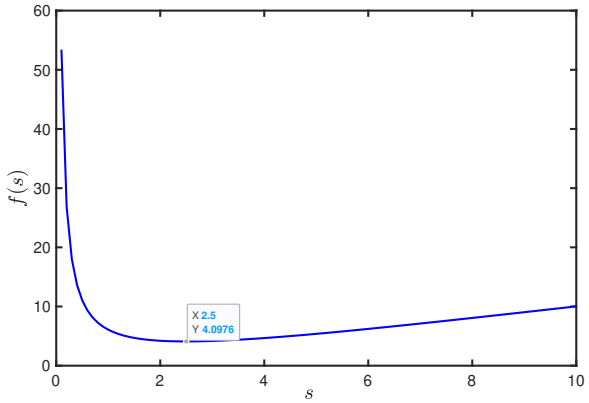


FIGURE A.13: Find s for $\min f(s)$.

From numerical experiments shown in Fig. A.13, we conclude that we can get $\min f(s)$ and minimum $\max_{x[t_j]\in[-1,1]} \mathrm{Var}[Y[t_j]]$, e.g., $k \approx \frac{\epsilon}{2.5}$ when $s \approx 2.5$.

(II) For Three-Outputs, the variance of $Y[t_j]$ is

$$\begin{aligned}\mathrm{Var}[Y[t_j]] &= \frac{d}{k}\left(\frac{(1-a)e^{\frac{2\epsilon}{k}}(e^{\frac{\epsilon}{k}}+1)^2}{(e^{\frac{\epsilon}{k}}-1)^2(e^{\frac{\epsilon}{k}}-a)^2}+\frac{b|x[t_j]|e^{\frac{2\epsilon}{k}}(e^{\frac{\epsilon}{k}}+1)^2}{(e^{\frac{\epsilon}{k}}-1)^2(e^{\frac{\epsilon}{k}}-a)^2}\right)-(x[t_j])^2\\ &= \frac{d}{\epsilon}\cdot s\cdot\left(\frac{(1-a)e^{\frac{2\epsilon}{k}}(e^{\frac{\epsilon}{k}}+1)^2}{(e^{\frac{\epsilon}{k}}-1)^2(e^{\frac{\epsilon}{k}}-a)^2}+\frac{b|x[t_j]|e^{\frac{2\epsilon}{k}}(e^{\frac{\epsilon}{k}}+1)^2}{(e^{\frac{\epsilon}{k}}-1)^2(e^{\frac{\epsilon}{k}}-a)^2}\right)-(x[t_j])^2,\end{aligned}$$

where b is from Eq. (A.23) and a is from Eq. (A.22).

Let $x[t_j]' = \frac{dbe^{\frac{2\epsilon}{k}}(e^{\frac{\epsilon}{k}}+1)^2}{2k(e^{\frac{\epsilon}{k}}-1)^2(e^{\frac{\epsilon}{k}}-a)^2}$, if $0 < x[t_j]' < 1$, the worst-case noise variance of Y is

$$\max_{x[t_j]\in[-1,1]}\mathrm{Var}[Y[t_j]] = \begin{cases}\mathrm{Var}[x[t_j]'], & \text{if } 0 < x[t_j]' < 1,\\ \max\{\mathrm{Var}[0],\mathrm{Var}[1]\}, & \text{otherwise.}\end{cases} \tag{A.148}$$

Let $s = \frac{\epsilon}{k}$, and then we have

$$x[t_j]' = \frac{d}{\epsilon}\cdot s\frac{be^{2s}(e^s+1)^2}{2(e^s-1)^2(e^s-a)^2} = \frac{d}{\epsilon}\cdot s\frac{ae^s(e^s+1)^2}{2(e^s-1)(e^s-a)^2}.$$

If $0 < \frac{d}{\epsilon}\cdot s\frac{ae^s(e^s+1)^2}{2(e^s-1)(e^s-a)^2} < 1$,

$$\begin{aligned}&\max_{x[t_j]\in[-1,1]}\mathrm{Var}[Y[t_j]] = \max_{x[t_j]\in[-1,1]}\mathrm{Var}[x[t_j]'] =\\ &= \frac{d}{k}\left(\frac{(1-a)e^{\frac{2\epsilon}{k}}(e^{\frac{\epsilon}{k}}+1)^2}{(e^{\frac{\epsilon}{k}}-1)^2(e^{\frac{\epsilon}{k}}-a)^2}+\frac{bx[t_j]'e^{\frac{2\epsilon}{k}}(e^{\frac{\epsilon}{k}}+1)^2}{(e^{\frac{\epsilon}{k}}-1)^2(e^{\frac{\epsilon}{k}}-a)^2}\right)-(x[t_j]')^2\\ &= \frac{d^2}{\epsilon^2}\cdot s^2\frac{b^2e^{4s}(e^s+1)^4}{2(e^s-1)^4(e^s-a)^4}-\frac{d^2}{\epsilon^2}\cdot s^2\frac{b^2e^{4s}(e^s+1)^4}{4(e^s-1)^4(e^s-a)^4}\\ &\quad+\frac{d}{\epsilon}\cdot s\frac{(1-a)e^{2s}(e^s+1)^2}{(e^s-1)^2(e^s-a)^2}\\ &= \frac{d^2}{\epsilon^2}\cdot s^2\frac{b^2e^{4s}(e^s+1)^4}{4(e^s-1)^4(e^s-a)^4}+\frac{d}{\epsilon}\cdot s\frac{(1-a)e^{2s}(e^s+1)^2}{(e^s-1)^2(e^s-a)^2}.\end{aligned} \tag{A.149}$$

Substituting $b = a\cdot\frac{e^s-1}{e^s}$ into Eq. (A.149) yields

$$\frac{d^2}{\epsilon^2}\cdot s^2\frac{a^2e^{2s}(e^s+1)^4}{4(e^s-1)^2(e^s-a)^4}+\frac{d}{\epsilon}\cdot s\frac{(1-a)e^{2s}(e^s+1)^2}{(e^s-1)^2(e^s-a)^2}.$$

– If $\epsilon < \ln 2$, $a = 0$ and $b = 0$, so that the first-order derivative of $\max_{x[t_j]\in[-1,1]} \mathrm{Var}[Y[t_j]]$ is

$$\max_{x[t_j]\in[-1,1]} \mathrm{Var}[Y[t_j]]' = \frac{d}{\epsilon} \cdot \frac{(e^s+1)(-4se^s+e^{2s}-1)}{(e^s-1)^3}. \tag{A.150}$$

When $\max_{x[t_j]\in[-1,1]} \mathrm{Var}[Y[t_j]]' = 0$, we have root $s \approx 2.18$.

– If $\ln 2 < \epsilon < \ln 5.5$, by using numerical experiments, we have optimal $s \approx 2.5$.

– If $\epsilon \geq \ln 5.5$, by using numerical experiments, we have optimal $s \approx 2.5$.

Therefore, we pick $s \approx 2.5$, i.e., $k \approx \frac{\epsilon}{2.5}$, for simplicity.

■

A.21 Extending `Three-Outputs` for Multiple Numeric Attributes

Lemma 15. The variance of $Y[t_j]$ induced by `Three-Outputs` is

$$\mathrm{Var}[Y[t_j]] = \frac{d}{k}\left(\frac{(1-a)e^{\frac{2\epsilon}{k}}(e^{\frac{\epsilon}{k}}+1)^2}{(e^{\frac{\epsilon}{k}}-1)^2(e^{\frac{\epsilon}{k}}-a)^2} + \frac{b|x[t_j]|e^{\frac{2\epsilon}{k}}(e^{\frac{\epsilon}{k}}+1)^2}{(e^{\frac{\epsilon}{k}}-1)^2(e^{\frac{\epsilon}{k}}-a)^2}\right) - (x[t_j])^2,$$

where x is a numeric tuple with d dimensions perturbed as Y using ϵ-LDP, and each t_j has d attributes.

Proof. The variance of $Y[t_j]$ is computed as

$$\begin{aligned}\mathrm{Var}[Y[t_j]] &= \mathbb{E}[(Y[t_j])^2] - \mathbb{E}[Y[t_j]]^2 \\ &= \frac{k}{d}\mathbb{E}[(\frac{d}{k}y_j)^2] - (x[t_j])^2 \\ &= \frac{d}{k}\mathbb{E}[(y_j)^2] - (x[t_j])^2. \end{aligned} \tag{A.151}$$

Then, we use the variance in Eq. (A.113) to compute

$$\begin{aligned}\mathbb{E}[(y_j)^2] &= \mathrm{Var}[y_j] + (\mathbb{E}[y_j])^2 \\ &= \frac{(1-a)e^{\frac{2\epsilon}{k}}(e^{\frac{\epsilon}{k}}+1)^2}{(e^{\frac{\epsilon}{k}}-1)^2(e^{\frac{\epsilon}{k}}-a)^2} + \frac{b|x[t_j]|e^{\frac{2\epsilon}{k}}(e^{\frac{\epsilon}{k}}+1)^2}{(e^{\frac{\epsilon}{k}}-1)^2(e^{\frac{\epsilon}{k}}-a)^2} - (x[t_j])^2 + (x[t_j])^2 \\ &= \frac{(1-a)e^{\frac{2\epsilon}{k}}(e^{\frac{\epsilon}{k}}+1)^2}{(e^{\frac{\epsilon}{k}}-1)^2(e^{\frac{\epsilon}{k}}-a)^2} + \frac{b|x[t_j]|e^{\frac{2\epsilon}{k}}(e^{\frac{\epsilon}{k}}+1)^2}{(e^{\frac{\epsilon}{k}}-1)^2(e^{\frac{\epsilon}{k}}-a)^2}.\end{aligned}$$

Finally, we obtain

$$\mathrm{Var}[Y[t_j]] = \frac{d}{k}\left(\frac{(1-a)e^{\frac{2\epsilon}{k}}(e^{\frac{\epsilon}{k}}+1)^2}{(e^{\frac{\epsilon}{k}}-1)^2(e^{\frac{\epsilon}{k}}-a)^2} + \frac{b|x[t_j]|e^{\frac{2\epsilon}{k}}(e^{\frac{\epsilon}{k}}+1)^2}{(e^{\frac{\epsilon}{k}}-1)^2(e^{\frac{\epsilon}{k}}-a)^2}\right) - (x[t_j])^2.$$

□

Appendix B

Appendix for Chapter 4

B.1 Proof of Theorem 1

Proof. (i) As noted in the statement of Theorem 1, we suppose that before answering query Q_m and after answering $Q_1, Q_2, \ldots, Q_{m-1}$, the privacy loss $L_{\widetilde{Q}_1 \| \widetilde{Q}_2 \| \ldots \| \widetilde{Q}_{m-1}}(D, D')$ is given by $\mathcal{N}(\frac{\mathcal{A}(D,D')}{2}, \mathcal{A}(D, D'))$ for some $\mathcal{A}(D, D')$. Later we will show the existence of such $\mathcal{A}(D, D')$. Then, when y_i follows the probability distribution of random variable $\widetilde{Q}_i(D)$ for each $i \in \{1, 2, \ldots, m-1\}$, we have the following for the privacy loss $L_{\widetilde{Q}_1 \| \widetilde{Q}_2 \| \ldots \| \widetilde{Q}_{m-1}}(D, D'; y_1, y_2, \ldots, y_{m-1})$:

$$\begin{aligned} &L_{\widetilde{Q}_1 \| \widetilde{Q}_2 \| \ldots \| \widetilde{Q}_{m-1}}(D, D'; y_1, y_2, \ldots, y_{m-1}) \\ &:= \ln \frac{\mathbb{F}\left[\cap_{i=1}^{m-1}\left[\widetilde{Q}_i(D) = y_i\right]\right]}{\mathbb{F}\left[\cap_{i=1}^{m-1}\left[\widetilde{Q}_i(D') = y_i\right]\right]} \sim \mathcal{N}\left(\frac{\mathcal{A}(D, D')}{2}, \mathcal{A}(D, D')\right), \end{aligned} \tag{B.1}$$

where we use "$\sim$" to mean "obeying the distribution".

Now, we need to analyze the privacy loss after answering the m queries $Q_1, Q_2, \ldots, Q_m$. We look at the privacy loss $L_{\widetilde{Q}_1 \| \widetilde{Q}_2 \| \ldots \| \widetilde{Q}_m}(D, D'; y_1, y_2, \ldots, y_m)$ defined as follows:

$$L_{\widetilde{Q}_1 \| \widetilde{Q}_2 \| \ldots \| \widetilde{Q}_m}(D, D'; y_1, y_2, \ldots, y_m) := \ln \frac{\mathbb{F}\left[\cap_{i=1}^{m}\left[\widetilde{Q}_i(D) = y_i\right]\right]}{\mathbb{F}\left[\cap_{i=1}^{m}\left[\widetilde{Q}_i(D') = y_i\right]\right]}. \tag{B.2}$$

Hence, we use (B.1) to analyze (B.2). From (4.2), since $\widetilde{Q}_m(D)$ is generated by reusing $\widetilde{Q}_j(D)$ and generating additional noise (if necessary), where j is an integer in $\{1, 2, \ldots, m-1\}$ as noted in the statement of Theorem 1, we have

$$\begin{aligned}
&L_{\widetilde{Q}_1 \| \widetilde{Q}_2 \| \ldots \| \widetilde{Q}_m}(D, D'; y_1, y_2, \ldots, y_m) \\
&= \ln \frac{\mathbb{F}\left[\cap_{i=1}^{m-1}\left[\widetilde{Q}_i(D) = y_i\right]\right] \mathbb{F}\left[\widetilde{Q}_m(D) = y_m \mid \widetilde{Q}_j(D) = y_j\right]}{\mathbb{F}\left[\cap_{i=1}^{m-1}\left[\widetilde{Q}_i(D') = y_i\right]\right] \mathbb{F}\left[\widetilde{Q}_m(D') = y_m \mid \widetilde{Q}_j(D') = y_j\right]} \\
&= \ln \frac{\mathbb{F}\left[\cap_{i=1}^{m-1}\left[\widetilde{Q}_i(D) = y_i\right]\right]}{\mathbb{F}\left[\cap_{i=1}^{m-1}\left[\widetilde{Q}_i(D') = y_i\right]\right]} + \ln \frac{\mathbb{F}\left[\widetilde{Q}_m(D) = y_m \mid \widetilde{Q}_j(D) = y_j\right]}{\mathbb{F}\left[\widetilde{Q}_m(D') = y_m \mid \widetilde{Q}_j(D') = y_j\right]}.
\end{aligned} \tag{B.3}$$

We now discuss the first term $\ln \frac{\mathbb{F}\left[\cap_{i=1}^{m-1}\left[\widetilde{Q}_i(D)=y_i\right]\right]}{\mathbb{F}\left[\cap_{i=1}^{m-1}\left[\widetilde{Q}_i(D')=y_i\right]\right]}$ and the second term $\ln \frac{\mathbb{F}\left[\widetilde{Q}_m(D)=y_m \mid \widetilde{Q}_j(D)=y_j\right]}{\mathbb{F}\left[\widetilde{Q}_m(D')=y_m \mid \widetilde{Q}_j(D')=y_j\right]}$ in the last row of (B.3). To begin with, from (B.1), the first term in the last row of (B.3) follows the Gaussian distribution $\mathcal{N}(\frac{\mathcal{A}(D,D')}{2}, \mathcal{A}(D, D'))$. Next, we analyze the second term in the last row of (B.3).

When $\widetilde{Q}_j(D)$ and $\widetilde{Q}_m(D)$ take y_j and y_m respectively, $\widetilde{Q}_j(D) - Q_j(D)$ and $\widetilde{Q}_m(D) - Q_m(D) - r[\widetilde{Q}_j(D) - Q_j(D)]$ take the following defined g_j and g_m respectively:

$$g_j := y_j - Q_j(D), \tag{B.4}$$

$$g_m := y_m - Q_m(D) - r[y_j - Q_j(D)]. \tag{B.5}$$

For D' being a neighboring dataset of D, we further define

$$h_j := Q_j(D) - Q_j(D'), \tag{B.6}$$

$$h_m := Q_m(D) - Q_m(D'), \tag{B.7}$$

so that

$$g_j + h_j = y_j - Q_j(D'), \tag{B.8}$$

$$g_m + h_m - rh_j = y_m - Q_m(D') - r[y_j - Q_j(D')]. \tag{B.9}$$

Note that h_j and h_m are the same since Q_j and Q_m are the same. From the above analysis, we obtain :

$$\begin{aligned}
&\mathbb{F}\left[\widetilde{Q}_m(D) = y_m \mid \widetilde{Q}_j(D) = y_j\right] \\
&= \mathbb{F}\left[\begin{array}{l} \widetilde{Q}_m(D) - Q_m(D) - r[\widetilde{Q}_j(D) - Q_j(D)] = g_m \\ \mid \widetilde{Q}_j(D) = y_j \end{array}\right] \\
&\overset{\text{(b)}}{=} \frac{1}{\sqrt{2\pi({\sigma_m}^2 - r^2{\sigma_j}^2)}} e^{-\frac{{g_m}^2}{2({\sigma_m}^2 - r^2{\sigma_j}^2)}},
\end{aligned} \tag{B.10}$$

where step (b) follows since where $\widetilde{Q}_j(D) - Q_j(D)$ is a zero-mean Gaussian random variable with variance σ_j^2 and $\widetilde{Q}_m(D) - Q_m(D) - r[\widetilde{Q}_j(D) - Q_j(D)]$ is a zero-mean Gaussian random variable with variance ${\sigma_m}^2 - r^2{\sigma_j}^2$.

Similarly, for dataset D', we have :

$$\begin{aligned}
&\mathbb{F}\left[\widetilde{Q}_m(D') = y_m \mid \widetilde{Q}_j(D') = y_j\right] \\
&= \mathbb{F}\left[\begin{array}{l} \widetilde{Q}_m(D') - Q_m(D') - r[\widetilde{Q}_j(D') - Q_j(D')] \\ = g_m + h_m - rh_j \\ \mid \widetilde{Q}_j(D') = y_j \end{array}\right] \\
&\overset{\text{(b)}}{=} \frac{1}{\sqrt{2\pi({\sigma_m}^2 - r^2{\sigma_j}^2)}} e^{-\frac{(g_m + h_m - rh_j)^2}{2({\sigma_m}^2 - r^2{\sigma_j}^2)}},
\end{aligned} \tag{B.11}$$

where step (b) follows since where $\widetilde{Q}_j(D') - Q_j(D')$ is a Gaussian random variable with variance σ_j^2 and $\widetilde{Q}_m(D') - Q_m(D') - r[\widetilde{Q}_j(D') - Q_j(D')$ is a zero-mean Gaussian random variable with variance ${\sigma_m}^2 - r^2{\sigma_j}^2$.

Then, we obtain:

$$\begin{aligned}
&\ln \frac{\mathbb{F}\left[\widetilde{Q}_m(D) = y_m \mid \widetilde{Q}_j(D) = y_j\right]}{\mathbb{F}\left[\widetilde{Q}_m(D') = y_m \mid \widetilde{Q}_j(D') = y_j\right]} \\
&= \ln \frac{\frac{1}{\sqrt{2\pi({\sigma_m}^2 - r^2{\sigma_j}^2)}} e^{-\frac{{g_m}^2}{2({\sigma_m}^2 - r^2{\sigma_j}^2)}}}{\frac{1}{\sqrt{2\pi({\sigma_m}^2 - r^2{\sigma_j}^2)}} e^{-\frac{(g_m + h_m - rh_j)^2}{2({\sigma_m}^2 - r^2{\sigma_j}^2)}}} \\
&= \frac{(g_m + h_m - rh_j)^2 - {g_m}^2}{2({\sigma_m}^2 - r^2{\sigma_j}^2)} \\
&= \frac{g_m(h_m - rh_j)}{{\sigma_m}^2 - r^2{\sigma_j}^2} + \frac{(h_m - rh_j)^2}{2({\sigma_m}^2 - r^2{\sigma_j}^2)}.
\end{aligned} \tag{B.12}$$

The above (B.12) presents the second term in the last row of (B.3). At first glance, it may seem that the first term $\ln \frac{\mathbb{F}\left[\cap_{i=1}^{m-1}\left[\widetilde{Q}_i(D)=y_i\right]\right]}{\mathbb{F}\left[\cap_{i=1}^{m-1}\left[\widetilde{Q}_i(D')=y_i\right]\right]}$ and the second term $\ln \frac{\mathbb{F}\left[\widetilde{Q}_m(D)=y_m \mid \widetilde{Q}_j(D)=y_j\right]}{\mathbb{F}\left[\widetilde{Q}_m(D')=y_m \mid \widetilde{Q}_j(D')=y_j\right]}$ in the last row of (B.3) are dependent since they both involve y_j. However, we have shown from (B.12) above that the second term in the last row of (B.3) depends on only the random variable g_m (note that terms in (B.12) other than g_m are all given), which is the amount of additional Gaussian noise used to generated $\widetilde{Q}_m(D)$ according to (4.2) and (B.5); i.e., the second term in the last row of (B.3) is actually independent of the first term in the last row of (B.3). From (B.1), the first term in the last row of (B.3) follows the Gaussian distribution $\mathcal{N}(\frac{\mathcal{A}(D,D')}{2}, \mathcal{A}(D,D'))$. Next, we show that (B.12) presenting the second term in the last row of (B.3) also follows a Gaussian distribution.

Since g_m follows a zero-mean Gaussian distribution with variance ${\sigma_m}^2 - r^2{\sigma_j}^2$, clearly $\frac{g_m(h_m - rh_j)}{{\sigma_m}^2 - r^2{\sigma_j}^2}$ follows a zero-mean Gaussian distribution with variance given by

$$\left[\frac{(h_m - rh_j)}{{\sigma_m}^2 - r^2{\sigma_j}^2}\right]^2 \times ({\sigma_m}^2 - r^2{\sigma_j}^2) = \frac{(h_m - rh_j)^2}{{\sigma_m}^2 - r^2{\sigma_j}^2}. \tag{B.13}$$

Since Q_m and Q_j are the same, we obtain from Eq. (B.6) and Eq. (B.7) that $h_j = h_m = Q_m(D) - Q_m(D')$, which we use to write Eq. (B.13) as

$$\frac{[\|Q_m(D) - Q_m(D')\|_2]^2(1-r)^2}{{\sigma_m}^2 - r^2{\sigma_j}^2}. \tag{B.14}$$

Summarizing the above, privacy loss is

$$B_r(D, D') := \mathcal{A}(D, D') + \frac{[\|Q_m(D) - Q_m(D')\|_2]^2(1-r)^2}{{\sigma_m}^2 - r^2{\sigma_j}^2}. \tag{B.15}$$

As noted in the statement of Theorem 1, we suppose that before answering query Q_m and after answering $Q_1, Q_2, \ldots, Q_{m-1}$, the privacy loss $L_{\widetilde{Q}_1\|\widetilde{Q}_2\|\cdots\|\widetilde{Q}_{m-1}}(D, D')$ is given by $\mathcal{N}(\frac{\mathcal{A}(D,D')}{2}, \mathcal{A}(D, D'))$ for some $\mathcal{A}(D, D')$. With the above result (B.15), we can actually show that there indeed exists such $\mathcal{A}(D, D')$. This follows from mathematical induction. For the base case; i.e., when only one query is answered, the result follows from Lemma 3 of [173]. The induction step is given by the above result (B.15). Hence, we have shown the existence of $\mathcal{A}(D, D')$. With this result and (B.15), we have completed proving Result (i) of Theorem 1.

(ii) The optimal r is obtained by minimizing $B_r(D, D')$ and hence minimizing $\frac{(1-r)^2}{{\sigma_m}^2 - r^2{\sigma_j}^2}$. Analyzing the monotonicity of this expression, we derive the optimal r as in Eq. (4.3). The first-order derivative of $B_r(D, D')$ to r is:

$$B_r(D, D')' = \frac{-2(r{\sigma_j}^2 - {\sigma_m}^2)(r-1)}{(r^2{\sigma_j}^2 - {\sigma_m}^2)^2}. \tag{B.16}$$

- Case 1: if $\sigma_m \geq \sigma_j$, $B_r(D, D')' \geq 0$ when $r \in [1, \frac{\sigma_m}{\sigma_j}]$, and $B_r(D, D')' < 0$ when $r \in (-\infty, 1) \cup (\frac{\sigma_m}{\sigma_j}, +\infty)$. Hence, the optimal r to minimize $B_r(D, D')$ is at $r = 1$.
- Case 2: if $\sigma_m < \sigma_j$, $B_r(D, D')' \geq 0$ when $r \in [\frac{\sigma_m}{\sigma_j}, 1]$, and $B_r(D, D')' < 0$ when $r \in (-\infty, \frac{\sigma_m}{\sigma_j}) \cup (1, +\infty)$. Hence, the optimal r to minimize $B_r(D, D')$ is at $r = (\frac{\sigma_m}{\sigma_j})^2$.

Thus, we obtain optimal values of r as Eq. (4.3). □

B.2 Proof of Lemma 14

Proof. Consider a query R with ℓ_2-sensitivity being 1. Let $\widetilde{R}$ be the mechanism of adding Gaussian noise amount $\mu := \frac{1}{\sqrt{\max_{\text{neighboring datasets } D,\, D'} V(D,D')}}$ to R. From Corollary 1, the privacy loss of randomized mechanism $\widetilde{R}$ with respect to

neighboring datasets D and D' is given by $\mathcal{N}(\frac{U(D,D')}{2}, U(D,D'))$ for $U(D,D') := \frac{[\|R(D)-R(D')\|_2]^2}{\mu^2}$. By considering the ℓ_2-sensitivity of R (i.e., $\|R(D)-R(D')\|_2$) as 1, $\max_{\text{neighboring datasets } D,D'} V(D,D')$ and $\max_{\text{neighboring datasets } D,D'} U(D,D')$ are the same. In addition, from Theorem 5 of [173], letting Y (resp., $\widetilde{R}$) satisfy (ϵ,δ)-differential privacy can be converted to a condition on $\max_{\text{neighboring datasets } D,D'} V(D,D')$ (resp., $\max_{\text{neighboring datasets } D,D'} U(D,D')$). Then letting Y satisfy (ϵ,δ)-differential privacy is the same as letting $\widetilde{R}$ satisfy (ϵ,δ)-differential privacy. From Lemma 2.1, $\widetilde{R}$ achieves (ϵ,δ)-differential privacy with $\mu = \text{Gaussian}(1,\epsilon,\delta)$; i.e., if $\max_{\text{neighboring datasets } D,D'} V(D,D') = [\text{Gaussian}(1,\epsilon,\delta)]^{-2}$. Summarizing the above, we complete proving Lemma 14. □

B.3 Proof of Theorem 2

Proof. We use Theorem 1 to show Results ① ② and ③ of Theorem 2. Proof of ①: In Case 2A) and Case 2C), Q_m can reuse previous noise. Hence, the privacy loss will still be $\mathcal{N}(\frac{\mathcal{A}(D,D')}{2}, \mathcal{A}(D,D'))$ according to Eq. (4.4).
Proof of ②: In Case 1), Q_m cannot reuse previous noisy answers, and the new noise follows $\mathcal{N}(0,\sigma_m)$. Thus, $B(D,D') := \mathcal{A}(D,D') + \frac{[\|Q_m(D)-Q_m(D')\|_2]^2}{{\sigma_m}^2}$.
Proof of ③: In Case 2B), Q_m can reuse previous noisy answers partially, so we can prove it using Eq. (4.4).

Then, Lemma 14 further implies Results ④ ⑤ and ⑥ of Theorem 2.
Proof of ④: Q_m can fully reuse the old noisy result in Cases 2A) and 2C). Thus, the privacy level does not change.
Proof of ⑤: From Lemma 14, we have

$$\max_{\text{neighboring datasets } D,D'} \mathcal{A}(D,D') = [\text{Gaussian}(1,\epsilon_{\text{old}},\delta_{\text{old}})]^{-2}$$

and $\max_{\text{neighboring datasets } D,\, D'} \Big\{ \mathcal{A}(D, D') + [\|Q_m(D) - Q_m(D')\|_2]^2 \times \frac{1}{{\sigma_m}^2} \Big\} = [\text{Gaussian}(1, \epsilon_{\text{new}}, \delta_{\text{new}})]^{-2}$. The above two equations yield

$$\begin{aligned} &[\text{Gaussian}(1, \epsilon_{\text{new}}, \delta_{\text{new}})]^{-2} - [\text{Gaussian}(1, \epsilon_{\text{old}}, \delta_{\text{old}})]^{-2} \\ &= \max_{\text{neighboring datasets D,D'}} [\|Q_m(D) - Q_m(D')\|_2]^2 \times \frac{1}{{\sigma_m}^2} = {\Delta_{Q_m}}^2 \times \frac{1}{{\sigma_m}^2}. \end{aligned} \tag{B.17}$$

Hence, $\text{Gaussian}(\Delta_{Q_m}, \sqrt{\epsilon\_\text{squared}\_\text{cost}}, \delta_{\text{cost}}) = \sigma_m$.

Proof of ⑥: From Lemma 14, we have

$$\max_{\text{neighboring datasets } D,\, D'} \mathcal{A}(D, D') = [\text{Gaussian}(1, \epsilon_{\text{old}}, \delta_{\text{old}})]^{-2},$$

and $\max_{\text{neighboring datasets } D,\, D'} \Big\{ \mathcal{A}(D, D') + [\|Q_m(D) - Q_m(D')\|_2]^2 \times \Big[\frac{1}{{\sigma_m}^2} - \frac{1}{[\min(\mathbf{\Sigma}_t)]^2} \Big] \Big\}$ $= [\text{Gaussian}(1, \epsilon_{\text{new}}, \delta_{\text{new}})]^{-2}$. The above two equations yield

$$\begin{aligned} &[\text{Gaussian}(1, \epsilon_{\text{new}}, \delta_{\text{new}})]^{-2} - [\text{Gaussian}(1, \epsilon_{\text{old}}, \delta_{\text{old}})]^{-2} \\ &= \max_{\text{neighboring datasets } D,\, D'} [\|Q_m(D) - Q_m(D')\|_2]^2 \times \left[\frac{1}{{\sigma_m}^2} - \frac{1}{[\min(\mathbf{\Sigma}_t)]^2} \right] \\ &= {\Delta_{Q_m}}^2 \times \left[\frac{1}{{\sigma_m}^2} - \frac{1}{[\min(\mathbf{\Sigma}_t)]^2} \right]. \end{aligned}$$

Then, by using the expression of $\text{Gaussian}(\Delta_Q, \epsilon, \delta)$ from Lemma 2.1, we further obtain Result ⑥. □

B.4 Proof of Theorem 3

Proof. First, from Theorem 2, after Algorithm 7 is used to answer all n queries with query Q_i being answered under (ϵ_i, δ_i)-differential privacy, the total privacy loss with respect to neighboring datasets D and D' is given by $\mathcal{N}(\frac{G(D,D')}{2}, G(D, D'))$ for some $G(D, D')$.

Next, we use Theorem 2 to further show that the expression of $G(D, D')$ is given by Eq. (4.5). From Theorem 2, among all queries, only queries belonging to Cases 1) and 2B) contribute to $G(D, D')$. Below we discuss the contributions respectively.

With N_1 denoting the set of $i \in \{1, 2, \ldots, n\}$ such that Q_i is in Cases 1), we know from Result ② of Theorem 2 that the contributions of queries in Cases 1) to $G(D, D')$ is given by

$$\sum_{i \in N_1} \frac{[\|Q_i(D) - Q_i(D')\|_2]^2}{{\sigma_i}^2}. \tag{B.18}$$

Below we use Result ③ of Theorem 2 to compute the contributions of queries in Case 2B) to $G(D, D')$. For T_{2B} being the set of query types in Case 2B), we discuss each query type $t \in T_{2B}$ respectively.

From Result ③ of Theorem 2, the contribution to $G(D, D')$ by answering $Q_{j_{t,1}}$ under differential privacy is

$$[\|Q_{j_{t,1}}(D) - Q_{j_{t,1}}(D')\|_2]^2 \left(\frac{1}{\sigma^2_{j_{t,1}}} - \frac{1}{\sigma^2_{j_{t,0}}} \right).$$

Similarly, the contribution to $G(D, D')$ by answering $Q_{j_{t,2}}$ under differential privacy is

$$[\|Q_{j_{t,2}}(D) - Q_{j_{t,2}}(D')\|_2]^2 \left(\frac{1}{\sigma^2_{j_{t,2}}} - \frac{1}{\sigma^2_{j_{t,1}}} \right).$$

Similar analyses are repeated for additional type-t queries in Case 2B). In particular, for each $s \in \{1, 2, \ldots, m_t\}$, the contribution to $G(D, D')$ by answering $Q_{j_{t,s}}$ under differential privacy is

$$[\|Q_{j_{t,s}}(D) - Q_{j_{t,s}}(D')\|_2]^2 \left(\frac{1}{\sigma^2_{j_{t,s}}} - \frac{1}{\sigma^2_{j_{t,s-1}}} \right). \tag{B.19}$$

Summing all (B.19) for $s \in \{1, 2, \ldots, m_t\}$, we obtain that for each query type $t \in T_{2B}$, the contributions to $G(D, D')$ by answering $Q_{j_{t,1}}, Q_{j_{t,2}}, \ldots, Q_{j_{t,m_t}}$ under differential privacy is

$$\sum_{s \in \{1,2,\ldots,m_t\}} [\|Q_{j_{t,s}}(D) - Q_{j_{t,s}}(D')\|_2]^2 \left(\frac{1}{\sigma^2_{j_{t,s}}} - \frac{1}{\sigma^2_{j_{t,s-1}}} \right). \tag{B.20}$$

Since $Q_{j_{t,0}}, Q_{j_{t,1}}, \ldots, Q_{j_{t,m_t}}$ for $j_{t,0}, j_{t,1}, \ldots, j_{t,m_t}$ are all type-t queries, $\|Q_{j_{t,s}}(D) - Q_{j_{t,s}}(D')\|_2$ are all the same for $s \in \{1, 2, \ldots, m_t\}$. Hence, we write (B.20) as

$$\begin{aligned}
&\sum_{s\in\{1,2,\ldots,m_t\}} \left\{ \frac{[\|Q_{j_{t,s}}(D) - Q_{j_{t,s}}(D')\|_2]^2}{\sigma^2_{j_{t,s}}} \right. \\
&\qquad\qquad \left. - \frac{[\|Q_{j_{t,s-1}}(D) - Q_{j_{t,s-1}}(D')\|_2]^2}{\sigma^2_{j_{t,s-1}}} \right\} \\
&= \frac{[\|Q_{j_{t,m_t}}(D) - Q_{j_{t,m_t}}(D')\|_2]^2}{\sigma^2_{j_{t,m_t}}} \\
&\qquad - \frac{[\|Q_{j_{t,0}}(D) - Q_{j_{t,0}}(D')\|_2]^2}{\sigma^2_{j_{t,0}}}.
\end{aligned} \tag{B.21}$$

Summing all (B.21) for $t \in T_{2B}$, the contributions to $G(D, D')$ by answering all queries in Case 2B) is

$$\sum_{t\in T_{2B}} \left\{ \frac{[\|Q_{j_{t,m_t}}(D) - Q_{j_{t,m_t}}(D')\|_2]^2}{\sigma^2_{j_{t,m_t}}} - \frac{[\|Q_{j_{t,0}}(D) - Q_{j_{t,0}}(D')\|_2]^2}{\sigma^2_{j_{t,0}}} \right\}. \tag{B.22}$$

Then, $G(D, D')$ as the sum of (B.18) and (B.22) is given by Eq. (4.5).

By summarizing the above, we have proved that after Algorithm 7 is used to answer all n queries under differential privacy, and the total privacy loss with respect to neighboring datasets D and D' is given by $\mathcal{N}(\frac{G(D,D')}{2}, G(D, D'))$ for $G(D, D')$ in Eq. (4.5). Furthermore, under

$$\max_{\text{neighboring datasets } D,\, D'} \|Q_i(D) - Q_i(D')\|_2 = \Delta_{Q_i}$$

and

$$\begin{aligned}
&\max_{\text{neighboring datasets } D,\, D'} \|Q_{j_{t,m_t}}(D) - Q_{j_{t,m_t}}(D')\|_2 \\
&= \max_{\text{neighboring datasets } D,\, D'} \|Q_{j_{t,0}}(D) - Q_{j_{t,0}}(D')\|_2 \\
&= \Delta(\text{type-}t),
\end{aligned}$$

we use Eq. (4.5) to have $\max_{\text{neighboring datasets } D,\, D'} G(D, D')$ given by Eq. (4.6).

Finally, from Lemma 14, the total privacy cost of our Algorithm 7 can be given by $(\epsilon_{\text{ours}}, \delta_{\text{ours}})$-differential privacy for ϵ_{ours} satisfying

$$[\text{Gaussian}(1, \epsilon_{\text{ours}}, \delta_{\text{ours}})]^{-2} = \max_{\text{neighboring datasets } D,\, D'} G(D, D'),$$

or (ϵ, δ)-differential privacy for any ϵ and δ satisfying

$$[\text{Gaussian}(1, \epsilon, \delta)]^{-2} = \max_{\text{neighboring datasets D,D'}} G(D, D').$$

□

B.5 Utility of the Gaussian Mechanism

Proof. The noisy response for one-dimensional query Q_m is $\widetilde{Q}_m(D) = Q_m(D) + \mathcal{N}(0, \sigma^2)$. Letting the probability of $\|\widetilde{Q}_m(D) - Q_m(D)\|_p \leq \alpha$ be $1 - \beta$, and then we have

$$\begin{aligned}
1 - \beta &= \mathbb{P}\left[\|\widetilde{Q}_m(D) - Q_m(D)\|_p \leq \alpha\right] \\
&= \mathbb{P}\left[|\mathcal{N}(0, \sigma^2)| \leq \alpha\right] \\
&= \mathbb{P}\left[-\alpha \leq \mathcal{N}(0, \sigma^2) \leq \alpha\right] \\
&= \mathbb{P}\left[\mathcal{N}(0, \sigma^2) \leq \alpha\right] - \mathbb{P}\left[\mathcal{N}(0, \sigma^2) \leq -\alpha\right] \\
&= \frac{1}{2}\left[1 + \text{erf}\left(\frac{\alpha}{\sigma\sqrt{2}}\right)\right] - \frac{1}{2}\left[1 + \text{erf}\left(\frac{-\alpha}{\sigma\sqrt{2}}\right)\right] \\
&= \text{erf}\left(\frac{\alpha}{\sigma\sqrt{2}}\right), \qquad \text{(B.23)}
\end{aligned}$$

where $\text{erf}(\cdot)$ denotes the error function and the last step of Eq. (B.23) uses the fact that $\text{erf}(\cdot)$ is an odd function.

According to the two-sigma rule of Gaussian distribution [161], which can also be obtained from above equation that 95% values lie within two standard deviations of the mean. Thus, if we set $\alpha = 2\sigma$, $\beta \approx 0.05$. □

List of Author's Awards, Patents, and Publications[1]

Articles

1. **Local differential privacy based federated learning for the Internet of Things**
   Yang Zhao, Jun Zhao, Mengmeng Yang, Teng Wang, Ning Wang, Lingjuan Lyu, Dusit Niyato, Kwok-Yan Lam.
   IEEE Internet of Things Journal.

2. **A Blockchain-Based Approach for Saving and Tracking Differential-Privacy Cost**
   Yang Zhao, Jun Zhao, Jiawen Kang, Zehang Zhang, Dusit Niyato, Shuyu Shi, Kwok-Yan Lam.
   IEEE Internet of Things Journal.

3. **POSTER: Blockchain-Based Differential Privacy Cost Management System**
   Leong Mei Han*, **Yang Zhao***, and Jun Zhao.
   Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020.

4. **Privacy-Preserving Blockchain-Based Federated Learning for IoT Devices**
   Yang Zhao, Jun Zhao, Linshan Jiang, Rui Tan, Dusit Niyato, Zengxiang Li, Lingjuan Lyu, and Yingbo Liu.
   IEEE Internet of Things Journal.

[1]The superscript * indicates joint first authors

Bibliography

[1] Valentin Tudor, Vincenzo Gulisano, Magnus Almgren, and Marina Papatriantafilou. Bes: Differentially private event aggregation for large-scale IoT-based systems. *Future Generation Computer Systems*, 108:1241–1257, 2020. xi

[2] Muneeb Ul Hassan, Mubashir Husain Rehmani, and Jinjun Chen. Privacy preservation in blockchain based IoT systems: Integration issues, prospects, challenges, and future research directions. *Future Generation Computer Systems*, 97:512–529, 2019.

[3] Jinbo Xiong, Jun Ren, Lei Chen, Zhiqiang Yao, Mingwei Lin, Dapeng Wu, and Ben Niu. Enhancing privacy and availability for data clustering in intelligent electrical service of IoT. *IEEE Internet of Things Journal*, 6(2): 1530–1540, 2018.

[4] Jianqing Liu, Chi Zhang, and Yuguang Fang. Epic: A differential privacy framework to defend smart homes against Internet traffic analysis. *IEEE Internet of Things Journal*, 5(2):1206–1217, 2018.

[5] Keke Gai, Yulu Wu, Liehuang Zhu, Zijian Zhang, and Meikang Qiu. Differential privacy-based blockchain for industrial Internet-of-Things. *IEEE Transactions on Industrial Informatics*, 16(6):4156–4165, 2019. xi

[6] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *International Conference on the Theory and Applications of Cryptographic Techniques (Eurocrypt)*, pages 486–503, 2006. xi, 7, 15, 118

[7] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference (TCC)*, pages 265–284, 2006. xxiii, 6, 7, 14, 17, 27, 101, 102, 105

[8] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4): 211–407, 2014. xi, 3, 13, 15, 16, 36, 53, 79

[9] Cynthia Dwork and Guy Rothblum. Concentrated differential privacy. *arXiv preprint arXiv:1603.01887v1*, 2016. xi, 14

[10] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference (TCC)*, pages 635–658, 2016. 79

[11] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toni Pitassi, Omer Reingold, and Aaron Roth. Generalization in adaptive data analysis and holdout reuse. In *Conference on Neural Information Processing Systems (NIPS)*, pages 2350–2358, 2015. 14

[12] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 94–103, 2007. 54, 75

[13] Martín Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 308–318, 2016. xi, 77

[14] Jun Tang, Aleksandra Korolova, Xiaolong Bai, Xueqiang Wang, and Xiaofeng Wang. Privacy loss in Apple's implementation of differential privacy on macOS 10.12. *arXiv preprint arXiv:1709.02753*, 2017. xi, 69

[15] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067, 2014. xi

[16] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. In *Advances in Neural Information Processing Systems*, pages 3571–3580, 2017. xi

[17] John Duchi, Martin J Wainwright, and Michael I Jordan. Local privacy and minimax bounds: Sharp rates for probability estimation. In *Advances in Neural Information Processing Systems*, pages 1529–1537, 2013. xxiii, 6, 7, 17, 18

[18] Ning Wang, Xiaokui Xiao, Yin Yang, Jun Zhao, Siu Cheung Hui, Hyejin Shin, Junbum Shin, and Ge Yu. Collecting and analyzing multidimensional data with local differential privacy. In *IEEE International Conference on Data Engineering (ICDE)*, 2019. xxiii, 6, 7, 17, 18, 32, 46, 48, 53, 54, 55, 56, 87

[19] Raspberry Pi 4 tech specs, 2020. URL `https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/`. Accessed on August 13, 2020. xxiii, 110

[20] Kevin Ashton. That 'Internet of Things' thing. *RFID journal*, 22(7):97–114, 2009. 1

[21] Thiago H Silva, Pedro OS Vaz De Melo, Aline Carneiro Viana, Jussara M Almeida, Juliana Salles, and Antonio AF Loureiro. Traffic condition is more than colored lines on a map: characterization of waze alerts. In *International Conference on Social Informatics*, pages 309–318. Springer, 2013. 2

[22] Tobias Jeske. Floating car data from smartphones: What google and waze know about you and how hackers can control traffic. *Proceedings of the BlackHat Europe*, pages 1–12, 2013.

[23] Riri Fitri Sari, Adian Fatchur Rochim, Ellen Tangkudung, Arman Tan, and Timothy Marciano. Location-based mobile application software development: Review of waze and other apps. *Advanced Science Letters*, 23(3):2028–2032, 2017. 2

[24] Eduardo A Yamauchi, Patricia C de Souza, and Deógenes PS Junior. Prominent issues for privacy establishment in privacy policies of mobile apps. In *Proceedings of the 15th Brazilian Symposium on Human Factors in Computing Systems*, pages 1–9, 2016. 2

[25] Joshua Joy and Mario Gerla. Internet of vehicles and autonomous connected car-privacy and security issues. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9. IEEE, 2017.

[26] Trang Nguyen. Continuance intention in traffic-related social media: A privacy calculus perspective. *Journal of Internet Commerce*, pages 1–29, 2021.

[27] Yi Xu, Shuyue Wei, and Yansheng Wang. Privacy preserving online matching on ridesharing platforms. *Neurocomputing*, 406:371–377, 2020. 2

[28] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 111–125, 2008. 3

[29] IMDb. The Internet Movie Database., 2007. URL `http://www.imdb.com/`. 3

[30] Latanya Sweeney. k-Anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10 (05):557–570, 2002. 3

[31] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011. 3

[32] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282, 2017. 5, 7, 20, 23, 110

[33] Lingchen Zhao, Shengshan Hu, Qian Wang, Jianlin Jiang, Chao Shen, Xiangyang Luo, and Pengfei Hu. Shielding collaborative learning: Mitigating poisoning attacks through client-side detection. *IEEE Transactions on Dependable and Secure Computing*, PP(99):1–1, 10.1109/TDSC.2020.2986205, 2020. 5, 20, 28

[34] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(3):2031–2063, 2020. 5

[35] Jiawen Kang, Zehui Xiong, Dusit Niyato, Yuze Zou, Yang Zhang, and Mohsen Guizani. Reliable federated learning for mobile networks. *IEEE Wireless Communications*, 27(2):72–80, 2020. 5

[36] Yves Demazeau and J-P Müller. *Decentralized AI*. Elsevier, 1990. 5

[37] Jakub Konečnỳ, H. Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016. 5

[38] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the GAN: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 603–618, 2017. 6, 9, 33, 113

[39] Guowen Xu, Hongwei Li, Sen Liu, Kan Yang, and Xiaodong Lin. VerifyNet: Secure and verifiable federated learning. *IEEE Transactions on Information Forensics and Security*, 15:911–926, 2019. 6, 33

[40] Vincent Bindschaedler, Reza Shokri, and Carl A Gunter. Plausible deniability for privacy-preserving data synthesis. *Proceedings of the VLDB Endowment*, 10(5):481–492, 2017. 7

[41] Ryan Henry, Amir Herzberg, and Aniket Kate. Blockchain access privacy: Challenges and directions. *IEEE Security & Privacy*, 16(4):38–45, 2018. 8, 26

[42] Jiawen Kang, Zehui Xiong, Dusit Niyato, Shengli Xie, and Junshan Zhang. Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. *IEEE Internet of Things Journal*, 6(6):10700–10714, 2019. 8, 25

[43] Mu Yang, Andrea Margheri, Runshan Hu, and Vladimiro Sassone. Differentially private data sharing in a cloud federation with blockchain. *IEEE Cloud Computing*, 5(6):69–79, 2018. 8, 27, 28, 73

[44] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. When edge meets learning: Adaptive control for resource-constrained distributed machine learning. In *IEEE Conference on Computer Communications (INFOCOM)*, pages 63–71, 2018. 9, 11, 93

[45] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *IEEE Symposium on Security and Privacy (S&P)*, 2019. 9

[46] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. Mitigating sybils in federated learning poisoning. *arXiv preprint arXiv:1808.04866*, 2018. 10

[47] Yu Zhang, Tao Gu, and Xi Zhang. Mdldroid: a chainsgd-reduce approach to mobile deep learning for personal mobile sensing. In *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 73–84. IEEE, 2020. 10

[48] Karen Hao. How Apple personalizes Siri without hoovering up your data, 2019. URL `https://www.technologyreview.com/2019/12/11/131629/apple-ai-personalizes-siri-federated-learning/`. 10

[49] Juan Benet. IPFS-content addressed, versioned, P2P file system. *arXiv preprint arXiv:1407.3561*, 2014. 10, 25

[50] Xiaokui Xiao, Gabriel Bender, Michael Hay, and Johannes Gehrke. iReduct: Differential privacy with reduced relative errors. In *ACM International Conference on Management of Data (SIGMOD)*, pages 229–240, 2011. 16

[51] Chao Li and Gerome Miklau. An adaptive mechanism for accurate query answering under differential privacy. *Proceedings of the VLDB Endowment*, 5(6):514–525, 2012. 16

[52] Georgios Kellaris and Stavros Papadopoulos. Practical differential privacy via grouping and smoothing. *Proceedings of the VLDB Endowment*, 6(5): 301–312, 2013. 16

[53] Grigory Yaroslavtsev, Graham Cormode, Cecilia M Procopiuc, and Divesh Srivastava. Accurate and efficient private release of datacubes and contingency tables. In *IEEE International Conference on Data Engineering (ICDE)*, pages 745–756, 2013. 16

[54] Lu Ou, Zheng Qin, Shaolin Liao, Tao Li, and Dafang Zhang. Singular spectrum analysis for local differential privacy of classifications in the smart grid. *IEEE Internet of Things Journal*, 2020. 16, 17

[55] Wenjuan Tang, Ju Ren, Kun Deng, and Yaoxue Zhang. Secure data aggregation of lightweight e-healthcare IoT devices with fair incentives. *IEEE Internet of Things Journal*, 6(5):8714–8726, 2019.

[56] Meng Sun and Wee Peng Tay. On the relationship between inference and data privacy in decentralized IoT networks. *IEEE Transactions on Information Forensics and Security*, 15:852–866, 2019.

[57] Ping Zhao, Guanglin Zhang, Shaohua Wan, Gaoyang Liu, and Tariq Umer. A survey of local differential privacy for securing Internet of Vehicles. *The Journal of Supercomputing*, pages 1–22, 2019.

[58] Lin Sun, Jun Zhao, and Xiaojun Ye. Distributed clustering in the anonymized space with local differential privacy. *arXiv preprint arXiv:1906.11441*, 2019.

[59] Mehmet Emre Gursoy, Acar Tamersoy, Stacey Truex, Wenqi Wei, and Ling Liu. Secure and utility-aware data collection with condensed local differential privacy. *IEEE Transactions on Dependable and Secure Computing*, 2019.

[60] Soheila Ghane, Alireza Jolfaei, Lars Kulik, Kotagiri Ramamohanarao, and Deepak Puthal. Preserving privacy in the Internet of Connected Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[61] Lingjuan Lyu, Jiangshan Yu, Karthik Nandakumar, Yitong Li, Xingjun Ma, Jiong Jin, Han Yu, and Kee Siong Ng. Towards fair and privacy-preserving federated deep models. *IEEE Transactions on Parallel and Distributed Systems*, 31(11):2524–2541, 2020. 16, 17

[62] Lin Sun, Xiaojun Ye, Jun Zhao, Chenhui Lu, and Mengmeng Yang. Bisample: Bidirectional sampling for handling missing data with local differential privacy, 2020. 17

[63] John C Duchi, Michael I Jordan, and Martin J Wainwright. Minimax optimal procedures for locally private estimation. *Journal of the American Statistical Association*, 113(521):182–201, 2018. 19, 39, 44, 53, 54, 55, 56

[64] Chugui Xu, Ju Ren, Liang She, Yaoxue Zhang, Zhan Qin, and Kui Ren. EdgeSanitizer: Locally differentially private deep inference at the edge for mobile data analytics. *IEEE Internet of Things Journal*, 2019. 19

[65] Woo-Seok Choi, Matthew Tomei, Jose Rodrigo Sanchez Vicarte, Pavan Kumar Hanumolu, and Rakesh Kumar. Guaranteeing local differential privacy on ultra-low-power systems. In *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, pages 561–574. IEEE, 2018. 20

[66] Xiaofan He, Juan Liu, Richeng Jin, and Huaiyu Dai. Privacy-aware offloading in mobile-edge computing. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–6. IEEE, 2017. 20

[67] Xiong Li, Shanpeng Liu, Fan Wu, Saru Kumari, and Joel JPC Rodrigues. Privacy preserving data aggregation scheme for mobile edge computing assisted IoT applications. *IEEE Internet of Things Journal*, 2018. 20

[68] Pathum Chamikara Mahawaga Arachchige, Peter Bertok, Ibrahim Khalil, Dongxi Liu, Seyit Camtepe, and Mohammed Atiquzzaman. Local differential privacy for deep learning. *IEEE Internet of Things Journal*, 2019. 19, 20

[69] Vasyl Pihur. The podium mechanism: Improving on the laplace and staircase mechanisms. *arXiv preprint arXiv:1905.00191*, 2019. 20

[70] Jakub Konečnỳ, H. Brendan McMahan, and Daniel Ramage. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575*, 2015. 20

[71] Jianmin Chen, Xinghao Pan, Rajat Monga, Samy Bengio, and Rafal Jozefowicz. Revisiting distributed synchronous SGD. *arXiv preprint arXiv:1604.00981*, 2016. 20, 23, 34

[72] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020. 21

[73] Abhishek Bhowmick, John Duchi, Julien Freudiger, Gaurav Kapoor, and Ryan Rogers. Protection against reconstruction and its applications in private federated learning. *arXiv preprint arXiv:1812.00984*, 2018. 21

[74] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 691–706. IEEE, 2019. 21

[75] Ligeng Zhu and Song Han. Deep leakage from gradients. In *Federated Learning*, pages 17–31. Springer, 2020. 21

[76] Naman Agarwal, Ananda Theertha Suresh, Felix Yu, Sanjiv Kumar, and H. Brendan Mcmahan. cpSGD: Communication-efficient and differentially-private distributed SGD. *arXiv preprint arXiv:1805.10559*, 2018. 21

[77] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. iDLG: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020. 21

[78] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(3):2031–2063, 2020. 21, 22

[79] Meng Hao, Hongwei Li, Xizhao Luo, Guowen Xu, Haomiao Yang, and Sen Liu. Efficient and privacy-enhanced federated learning for industrial artificial intelligence. *IEEE Transactions on Industrial Informatics*, 2019. 22, 23

[80] Sidi Lu, Yongtao Yao, and Weisong Shi. Collaborative learning on the edges: A case study on connected vehicles. In *2nd USENIX Workshop on Hot Topics in Edge Computing (HotEdge 19)*, 2019. 21

[81] Romano Fantacci and Benedetta Picano. Federated learning framework for mobile edge computing networks. *CAAI Transactions on Intelligence Technology*, 5(1):15–21, 2020. 22

[82] Yuris Mulya Saputra, Dinh Thai Hoang, Diep N Nguyen, Eryk Dutkiewicz, Markus Dominik Mueck, and Srikathyayani Srikanteswara. Energy demand prediction with federated learning for electric vehicle networks. *arXiv preprint arXiv:1909.00907*, 2019. 21, 22

[83] Lingchen Zhao, Qian Wang, Qin Zou, Yan Zhang, and Yanjiao Chen. Privacy-preserving collaborative deep learning with unreliable participants. *IEEE Transactions on Information Forensics and Security*, 15(1):1486–1500, 2020. 22, 28

[84] Lingjuan Lyu, James C Bezdek, Xuanli He, and Jiong Jin. Fog-embedded deep learning for the Internet of Things. *IEEE Transactions on Industrial Informatics*, 2019. 22

[85] Linshan Jiang, Xin Lou, Rui Tan, and Jun Zhao. Differentially private collaborative learning for the IoT edge. In *International Workshop on Crowd Intelligence for Smart Cities: Technology and Applications (CISC)*, 2018. 22, 97, 101, 102, 105, 106

[86] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6):1205–1221, 2019. 22

[87] Xiaofei Wang, Yiwen Han, Chenyang Wang, Qiyang Zhao, Xu Chen, and Min Chen. In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning. *IEEE Network*, 33(5):156–165, 2019. 22

[88] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019. 22

[89] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[90] Takayuki Nishio and Ryo Yonetani. Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2019.

[91] Lingjuan Lyu, Han Yu, and Qiang Yang. Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133*, 2020. 22, 33

[92] Zaoxing Liu, Tian Li, Virginia Smith, and Vyas Sekar. Enhancing the privacy of federated learning with sketching. *arXiv preprint arXiv:1911.01812*, 2019. 22

[93] Koustabh Dolui, Illapha Cuba Gyllensten, Dietwig Lowet, Sam Michiels, Hans Hallez, and Danny Hughes. Poster: Towards privacy-preserving mobile applications with federated learning–the case of matrix factorization. In *The 17th Annual International Conference on Mobile Systems, Applications, and Services, Date: 2019/06/17-2019/06/21, Location: Seoul, Korea*, 2019. 22

[94] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 739–753. IEEE, 2019. 22

[95] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 2512–2520. IEEE, 2019. 22

[96] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. 2018. 22, 23, 36

[97] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pages 1–11, 2019. 22, 33

[98] Rui Hu, Yuanxiong Guo, Hongning Li, Qingqi Pei, and Yanmin Gong. Personalized federated learning with differential privacy. *IEEE Internet of Things Journal*, 2020. 23

[99] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. In *Advances in Neural Information Processing Systems*, pages 15453–15462, 2019.

[100] Aleksei Triastcyn and Boi Faltings. Federated learning with bayesian differential privacy. *arXiv preprint arXiv:1911.10071*, 2019. 23

[101] Yansheng Wang, Yongxin Tong, and Dingyuan Shi. Federated latent dirichlet allocation: A local differential privacy based framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6283–6290, 2020.

[102] Tian Li, Zaoxing Liu, Vyas Sekar, and Virginia Smith. Privacy for free: Communication-efficient learning with differential privacy using sketches. *arXiv preprint arXiv:1911.00972*, 2019. 22

[103] Andrew C Yao. Protocols for secure computations. In *Symposium on Foundations of Computer Science (FOCS)*, pages 160–164, 1982. 23

[104] Runhua Xu, Nathalie Baracaldo, Yi Zhou, Ali Anwar, and Heiko Ludwig. Hybridalpha: An efficient approach for privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pages 13–23, 2019. 23

[105] Xiling Li, Rafael Dowsley, and Martine De Cock. Privacy-preserving feature selection with secure multiparty computation. In *International Conference on Machine Learning*, pages 6326–6336. PMLR, 2021. 23

[106] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. {BatchCrypt}: Efficient homomorphic encryption for {Cross-Silo} federated learning. In *2020 USENIX Annual Technical Conference (USENIX ATC 20)*, pages 493–506, 2020. 23

[107] Fang-Jing Wu and Tie Luo. CrowdPrivacy: Publish more useful data with less privacy exposure in crowdsourced location-based services. *ACM Transactions on Privacy and Security (TOPS)*, 23(1):1–25, 2020. 24

[108] Tianqing Liang. Enabling privacy preservation and decentralization for attribute-based task assignment in crowdsourcing. *Journal of Computer and Communications*, 8(4):81–100, 2020. 24

[109] Yuanyuan He, Jianbing Ni, Ben Niu, Fenghua Li, and Xuemin Sherman Shen. Privbus: A privacy-enhanced crowdsourced bus service via fog computing. *Journal of Parallel and Distributed Computing*, 135:156–168, 2020. 24

[110] Jianhong Zhang, Qijia Zhang, and Shenglong Ji. A fog-assisted privacy-preserving task allocation in crowdsourcing. *IEEE Internet of Things Journal*, 2020.

[111] Ping Zhao, Haojun Huang, Xiaohui Zhao, and Daiyu Huang. P^3: Privacy-preserving scheme against poisoning attacks in mobile-edge computing. *IEEE Transactions on Computational Social Systems*, 2020. 24

[112] Jinliang Xu, Shangguang Wang, Bharat K Bhargava, and Fangchun Yang. A blockchain-enabled trustless crowd-intelligence ecosystem on mobile edge computing. *IEEE Transactions on Industrial Informatics*, 15(6):3538–3547, 2019.

[113] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, 2016.

[114] Weisong Shi and Schahram Dustdar. The promise of edge computing. *Computer*, 49(5):78–81, 2016. 24

[115] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. `https://bitcoin.org/bitcoin.pdf`, 2008. 24

[116] Investopedia. Blockchain. `https://www.investopedia.com/terms/b/blockchain.asp`, 2020. 25

[117] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 2014. 25

[118] Nick Szabo. Smart Contracts. `http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html`, 1994. 25

[119] Vitalik Buterin. A next-generation smart contract and decentralized application platform. *white paper*, 2014. 25

[120] Xiaoguang Li, Hui Li, Haonan Yan, Zelei Cheng, Wenhai Sun, and Hui Zhu. Mitigating query-flooding parameter duplication attack on regression models with high-dimensional gaussian mechanism, 2020. 25

[121] Jiawen Kang, Zehui Xiong, Dusit Niyato, Dongdong Ye, Dong In Kim, and Jun Zhao. Toward secure blockchain-enabled Internet of Vehicles: Optimizing consensus management using reputation and contract theory. *IEEE Transactions on Vehicular Technology*, 68(3):2906–2920, 2019. 26

[122] Jiawen Kang, Rong Yu, Xumin Huang, Maoqiang Wu, Sabita Maharjan, Shengli Xie, and Yan Zhang. Blockchain for secure and efficient data sharing in vehicular edge computing and networks. *IEEE Internet of Things Journal*, 6(3):4660–4670, 2018. 26

[123] Ya Che Tsai, Raylin Tso, Zi-Yuan Liu, and Kung Chen. An improved non-interactive zero-knowledge range proof for decentralized applications. In *2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON)*, pages 129–134. IEEE, 2019.

[124] Antonio Fernández Anta, Chryssis Georgiou, and Nicolas Nicolaou. Atomic appends: Selling cars and coordinating armies with multiple distributed ledgers. In *International Conference on Blockchain Economics, Security and Protocols (Tokenomics 2019)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020. 25

[125] Qi Feng, Debiao He, Sherali Zeadally, Muhammad Khurram Khan, and Neeraj Kumar. A survey on privacy protection in blockchain system. *Journal of Network and Computer Applications*, 126:45–58, 2019. 25

[126] Guy Zyskind, Oz Nathan, and Alex Pentland. Decentralizing privacy: Using blockchain to protect personal data. In *IEEE Security and Privacy Workshops (SPW)*, pages 180–184, 2015. 26

[127] Qi Xia, Emmanuel Boateng Sifah, Abla Smahi, Sandro Amofa, and Xiaosong Zhang. BBDS: Blockchain-based data sharing for electronic medical records in cloud environments. *Information*, 8(2):44, 2017. 26

[128] Yuan Lu, Qiang Tang, and Guiling Wang. Zebralancer: Private and anonymous crowdsourcing system atop open blockchain. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pages 853–865. IEEE, 2018. 26

[129] Shengshan Hu, Chengjun Cai, Qian Wang, Cong Wang, Xiangyang Luo, and Kui Ren. Searching an encrypted cloud meets blockchain: A decentralized, reliable and fair realization. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 792–800. IEEE, 2018. 26

[130] Matt Luongo and Corbin Pon. The Keep network: A privacy layer for public blockchains. Technical report, https://keep. network/whitepaper, 2017. 26

[131] Arthur Gervais, Ghassan Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. On the security and performance of proof of work blockchains. In *ACM SIGSAC Conference on Computer and Communication Security (CCS)*, 2016. 26

[132] Jordi Herrera-Joancomartí and Cristina Pérez-Solà. Privacy in bitcoin transactions: New challenges from blockchain scalability solutions. In *Modeling Decisions for Artificial Intelligence*, pages 26–44, 2016. 26

[133] Abubakar Sadiq Sani, Dong Yuan, Wei Bao, Phee Lep Yeoh, Zhao Yang Dong, Branka Vucetic, and Elisa Bertino. Xyreum: A high-performance and scalable blockchain for IIoT security and privacy. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 1920–1930. IEEE, 2019. 26

[134] Xidi Qu, Shengling Wang, Qin Hu, and Xiuzhen Cheng. Proof of federated learning: A novel energy-recycling consensus algorithm. *arXiv preprint arXiv:1912.11745*, 2019. 28

[135] Yunlong Lu, Xiaohong Huang, Yueyue Dai, Sabita Maharjan, and Yan Zhang. Blockchain and federated learning for privacy-preserved data sharing in Industrial IoT. *IEEE Transactions on Industrial Informatics*, 2019. 28

[136] Paritosh Ramanan, Kiyoshi Nakayama, and Ratnesh Sharma. Baffle: Blockchain based aggregator free federated learning. *arXiv preprint arXiv:1909.07452*, 2019.

[137] Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Blockchained on-device federated learning. *IEEE Communications Letters*, 2019.

[138] Bo Yin, Hao Yin, Yulei Wu, and Zexun Jiang. FDC: A secure federated deep learning mechanism for data collaborations in the Internet of Things. *IEEE Internet of Things Journal*, 2020.

[139] Sana Awan, Fengjun Li, Bo Luo, and Mei Liu. Poster: A reliable and accountable privacy-preserving federated learning framework using the blockchain. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 2561–2563, 2019. 28

[140] Jiasi Weng, Jian Weng, Jilian Zhang, Ming Li, Yue Zhang, and Weiqi Luo. DeepChain: Auditable and privacy-preserving deep learning with blockchain-based incentive. *IEEE Transactions on Dependable and Secure Computing*, 2019. 28

[141] Lingjuan Lyu, Jiangshan Yu, Karthik Nandakumar, Yitong Li, Xingjun Ma, and Jiong Jin. Towards fair and decentralized privacy-preserving deep learning with blockchain. *arXiv preprint arXiv:1906.01167*, 2019. 28

[142] Ming Li, Jian Weng, Anjia Yang, Wei Lu, Yue Zhang, Lin Hou, Jia-Nan Liu, Yang Xiang, and Robert Deng. CrowdBC: A blockchain-based decentralized framework for crowdsourcing. *IEEE Transactions on Parallel and Distributed Systems*, 2018. 28

[143] Teng Wang, Jun Zhao, Han Yu, Jinyan Liu, Xinyu Yang, Xuebin Ren, and Shuyu Shi. Privacy-preserving crowd-guided AI decision-making in ethical dilemmas. In *ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1311–1320, 2019. 34, 35

[144] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. Extremal mechanisms for local differential privacy. In *Advances in Neural Information Processing Systems*, pages 2879–2887, 2014. 39, 46

[145] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011. 55

[146] Steven Ruggles, Sarah Flood, Ronald Goeken, Josiah Grover, Erin Meyer, Jose Pacas, and Matthew Sobek. IPUMS USA: Version 10.0. *[dataset]. Minneapolis, MN: IPUMS. https://doi.org/10.18128/D010.V10.0*, 2020. 55

[147] Marco F Duarte and Yu Hen Hu. Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*, 64(7):826–838, 2004. 55, 58

[148] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*, 2019. 55, 58

[149] WWDC2016. June, 2016. Platforms State of the Union. `https://developer.apple.com/videos/play/wwdc2016/102/`. 69

[150] Leong Mei Han, Yang Zhao, and Jun Zhao. POSTER: Blockchain-based differential privacy cost management system. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, pages 925–927, 2020. 72

[151] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. *IEEE Transactions on Information Theory*, 63 (6):4037–4049, 2017. 77

[152] Ropsten's Official GitHub Page. `https://github.com/ethereum/ropsten`, accessed on 9 January 2019. 86

[153] David Sánchez, Josep Domingo-Ferrer, and Sergio Martínez. Improving the utility of differential privacy via univariate microaggregation. In *International Conference on Privacy in Statistical Databases*, pages 130–142. Springer, 2014. 87, 109

[154] Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to noninteractive database privacy. *Journal of the ACM (JACM)*, 60(2):1–25, 2013. 90

[155] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. Algorand: Scaling byzantine agreements for cryptocurrencies. In *ACM Symposium on Operating Systems Principles (SOSP)*, pages 51–68, 2017. 95, 98, 99

[156] Yunlong Mao, Shanhe Yi, Qun Li, Jinghao Feng, Fengyuan Xu, and Sheng Zhong. Learning from differentially private neural activations with edge computing. In *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 90–102. IEEE, 2018. 97

[157] Wenbo Wang, Dinh Thai Hoang, Peizhao Hu, Zehui Xiong, Dusit Niyato, Ping Wang, Yonggang Wen, and Dong In Kim. A survey on consensus mechanisms and mining strategy management in blockchain networks. *IEEE Access*, 7:22328–22370, 2019. 98

[158] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, pages 119–129, 2017. 98, 99

[159] Muhammad Shayan, Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. Biscotti: A ledger for private and secure peer-to-peer machine learning. *arXiv preprint arXiv:1811.09904*, 2018. 98, 99

[160] Yu Zhang and Mihaela Van der Schaar. Reputation-based incentive protocols in crowdsourcing applications. In *2012 Proceedings IEEE INFOCOM*, pages 2140–2148. IEEE, 2012. 99, 100

[161] Friedrich Pukelsheim. The three sigma rule. *The American Statistician*, 48 (2):88–91, 1994. 102, 184

[162] Han Yu, Zelei Liu, Yang Liu, Tianjian Chen, Mingshu Cong, Xi Weng, Dusit Niyato, and Qiang Yang. A fairness-aware incentive scheme for federated learning. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 393–399, 2020. 104

[163] Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database, 2010. URL `http://yann.lecun.com/exdb/mnist`. Accessed on March 1, 2019. 104, 110

[164] Dixing Xu, Mengyao Zheng, Linshan Jiang, Chaojie Gu, Rui Tan, and Peng Cheng. Lightweight and unobtrusive data obfuscation at IoT edge for remote inference. *IEEE Internet of Things Journal*, 2020. 105

[165] Mengyao Zheng, Dixing Xu, Linshan Jiang, Chaojie Gu, Rui Tan, and Peng Cheng. Challenges of privacy-preserving machine learning in IoT. In *Proceedings of the First International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things*, pages 1–7, 2019.

[166] Jed Mills, Jia Hu, and Geyong Min. Communication-efficient federated learning for wireless edge intelligence in IoT. *IEEE Internet of Things Journal*, 2019.

[167] Duo Liu, Chaoshu Yang, Shiming Li, Xianzhang Chen, Jinting Ren, Renping Liu, Moming Duan, Yujuan Tan, and Liang Liang. FitCNN: A cloud-assisted and low-cost framework for updating CNNs on IoT devices. *Future Generation Computer Systems*, 91:277–289, 2019.

[168] Florian Scheidegger, Luca Benini, Costas Bekas, and A Cristiano I Malossi. Constrained deep neural network architecture search for IoT devices accounting for hardware calibration. In *Advances in Neural Information Processing Systems*, pages 6054–6064, 2019.

[169] Atsutoshi Kumagai, Tomoharu Iwata, and Yasuhiro Fujiwara. Transfer anomaly detection by inferring latent domain representations. In *Advances in Neural Information Processing Systems*, pages 2467–2477, 2019. 105

[170] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, H. Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019. 110

[171] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020. 113

[172] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006. 118

[173] Borja Balle and Yu-Xiang Wang. Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning (ICML)*, 2018. 118, 179, 180