Privacy-Aware State Estimation based on Obfuscated Transformation and Differential Privacy

With applications to smart grids and supply chain economics

Lakshminarayanan Nandakumar

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Privacy-Aware State Estimation based on Obfuscated Transformation and Differential Privacy

With applications to smart grids and supply chain economics

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Lakshminarayanan Nandakumar

August 13, 2018

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Cover Image: Improving cloud computing security by Mark Warner, CC BY-ND. Taken from [1].

The work in this thesis was supported by the TU Delft Safety and Security Institute under the DSyS Grant. Their contribution is hereby gratefully acknowledged.

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Privacy-Aware State Estimation based on Obfuscated Transformation and Differential Privacy

by

Lakshminarayanan Nandakumar

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: August 13, 2018

Supervisor(s):

Dr.ir. Tamás Keviczky

Dr. Zekeriya Erkin

Gamze Tillem (PhD Candidate)

Reader(s):

Dr. Riccardo Ferrari

Vahab Rostampour (PhD Candidate)

Abstract

With the emergence of many modern automated systems around us that rely heavily on the private data collected from individuals, the problem of privacy-preserving data analysis is now gaining a significant attention in the field of systems and control. In this thesis, we investigate the privacy concerns of these systems arising in the process of state estimation - a well known and a widely studied concept in systems and control. Our work draws motivation from smart grids and supply chain economics, and hence, we study two different privacy problems in the context of state estimation and rely on cryptography to solve these challenges.

In the first problem, we study the privacy challenges of state estimation in smart grids. Smart grids promise a more reliable, efficient, economically viable, and an environment-friendly electricity infrastructure for the future. State estimation in smart grids plays a vital role in system monitoring, reliable operation, automation, and grid stabilization. However, the power consumption data collected from the users during estimation can be privacy-sensitive. Furthermore, the topology of the grid can be exploited by malicious entities during state estimation to launch attacks without getting detected. Motivated by the essence of a secure state estimation process, we propose a weighted-least-squares estimation carried out batch-wise at repeated intervals where the resource-constrained clients utilize a malicious cloud for computation services. We exploit a highly efficient and verifiable obfuscation-based cryptographic solution to perform the computations of the estimation process securely in the presence of a malicious adversary. Simulation results demonstrate a high level of obscurity both in time and frequency domain making it difficult for the malicious adversary to interpret information about the original power consumption data of the consumers and the grid topology from the obfuscated datasets.

Our second problem deals with the challenge of protecting a dynamical supply chain model while releasing the state sequences generated by the model for data aggregation to an external possible adversary. Releasing state samples generated by a dynamical system model with high accuracy for data aggregation and other statistical purposes can also be used for reverse engineering and estimating sensitive model parameters. Upon identification of the system model, the adversary may even use it for predicting sensitive data in the future. Hence, preserving a confidential dynamical process model is crucial for the survival of many industries. Motivated by the need to protect the system model as a trade secret, we propose a mechanism based on

differential privacy to render such model identification techniques ineffective while preserving the utility of the state samples for data aggregation purposes. We deploy differential privacy by generating noise according to the sensitivity of the query and adding it to the state vectors at each time instant. We derive analytical expressions to quantify the bound on the sensitivity function and estimate the minimum noise level required to guarantee differential privacy. Furthermore, we present numerical analysis and characterize the privacy-utility trade-off that arises when deploying differential privacy. Simulation results demonstrate that through differential privacy, we achieve acceptable privacy level sufficient to mislead the adversary while still managing to retain high utility level of the state samples for data aggregation.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I sincerely hope that this is the only section where I don't use any brains but all heart to pin down this incredible two-year journey at Delft. These two years have by far easily been the toughest journey of my life, and I would be lying if I say that I did it all alone by myself. My first thank you, love, and respect always goes to my Amma and Appa for being there for me - Period. Mom and Dad, you can finally cheer up as the wait is over; your only son is finally returning home soon after two long years. You mean the world to me.

Moving on to the professional front: I would first like to thank my supervisor Dr.ir. Tamás Keviczky without whom, this project would not have happened. I remember not having the best of times during the summer of 2017 (my only possible vacation time after a long dry hectic first year) trying to read a series of research papers to figure out a research problem for my thesis. It didn't happen, and soon after that, I lost all my confidence, and I even insisted on changing my project to a more control-oriented subject rather than having to deal with complex cryptology. But, it was Prof. Keviczky who showed trust and restored the confidence in me and motivated me to keep pursuing the original topic since it had good possibilities for exciting research work. Looking back now, I thoroughly enjoyed working on this inter-disciplinary project and have absolutely no regrets. So, thank you very much, sir, for not just that valuable advice, but for all your brilliant technical insights, fruitful discussions, and the time (especially for giving me appointments on short notice despite busy schedules) you spent with me during this project. Your constant reminder about the finiteness of time at every step in this project helped me stay pragmatic without getting carried away. I also thank you for offering me teaching assistantship for both of your courses and helping me win the DSyS Grant which, in turn, helped me support myself financially at this expensive place.

Next, I would like to thank my co-supervisors Dr. Zekeriya Erkin and Gamze Tillem for agreeing to collaborate with us on this project. Special thanks to Gamze for treating me more like your friend than your student, and for all the free coffee you bought me, whenever I visited your office. Your constant support and encouragement, especially during the tough times, was indeed helpful. Dr. Zeki, thank you for giving me time amidst your busy schedule. I especially thank you for your constructive criticisms which helped me improve myself as a research student, and insisting more on the motivational part behind my work which allowed me to maintain a global perspective of my research. I hope to have processed all your feedback and remarks during the writing of this thesis. Up next, I would like to thank Vahab for teaming up with me and helping me solve the second problem of my thesis with good clarity. I also take this opportunity to express my gratitude to Dr. Ricardo Ferrari for accepting to be a part of

my thesis defense committee and for your generosity in agreeing to have technical discussions with me whenever I was looking for new ideas, despite not being officially involved in this project. My thanks also extend to Riga Technical University for sharing real-time data with us for our simulations, and to Balijaa for helping me out with the MATPOWER software. Finally, a big word of thank you to Professor Pieter van Gelder, Professor Pieter Hartel, and the entire TU Delft Safety and Security Institute for believing in us and funding our project proposal.

To my friends at Delft: Arvind - Thanks bro for all your kindness and the care you have shown me during the last year. Naveen and Nivas - I will forever cherish our cooking sessions, late night discussions on cricket, politics, history, movies, and more importantly about life. Libardo - Thank you for being my team partner for all the first-year projects. You were such a pleasure to work with; countless assignments and deadlines have only made our friendship stronger despite some cross-cultural differences. Thanks to Sriram - our Norway trip and bike rides with Arvind proved to be a big stress-buster, Janani - for all the thesis-related discussions, and Navdeep. Also thanks to my wonderful senior and friend Sricharan for guiding me during the initial period at TU Delft. To all my friends back home; you know who you are. Thank You!

Last but not the least, I would like to thank the almighty for giving me the strength, courage, and determination to face the last two years, and blessing me with wonderful family and friends. I know this is just the beginning, and I foresee a long road ahead to making a real difference in the world. I will strive to accomplish it through my continued hard work, sincerity and dedication.

'Life has just begun...'

Delft, University of Technology
August 13, 2018

Lakshminarayanan (Srinath) Nandakumar

“ Information is powerful but it is how we use it that will define us.”
— *Zack Matere*

I dedicate this thesis to
My Amma and Appa

Chapter 1

Introduction

1-1 Motivation

In today's day and age, information is wealth since every information volunteered has a price tag attached to it. A growing concern in this digital era is the challenge raised by the emerging distributed automated systems around us, ranging from smart meters, buildings to intelligent transportation systems. These systems rely heavily on the private data collected from individuals for effective decision making and control [3]. However, the data collected from these private individuals can also be misused to breach their privacy. For example, with the installation of smart meters, fine-grained electricity measurements of households are recorded, and there are proven instances [4] where such measurements have revealed the nature of the electrical household activity, thereby compromising the consumer's privacy. Another example is the smart transportation service that requires traffic state estimates and forecasts, which in turn rely on the measurement of individual location traces. Similar concerns of data privacy arise in a variety of other areas, from supply chain economics to social networks and health care [3]. These problems are further amplified by the current trend within companies and government agencies to collect more information about private individuals. The recent Facebook privacy breach [5] where a personal data of 87m US voters were gathered and sold to a political consultancy Cambridge Analytica is one of the biggest example highlighting the problems of the current trend and the privacy concerns of the everyday user.

With systems becoming increasingly complex and larger in size, there is also a growing need to facilitate the distribution of decision-making process. While the success of the distributed systems relies on the power of information exchange, its fallibility lies in the power of information leakage [6]. Thus, privacy-preserving data analysis, once a topic actively discussed in the area of statistics and computer science [7], has now gained significant attention in systems and control. This thesis aims at exploring the privacy concerns of these information-driven systems arising in the process of state estimation - one of the most popular and widely studied concepts in systems and control.

State estimation is the process of estimating the *internal* states of a given real system from the given set of measurements of input and output. In many practical applications, all the

physical states of the system cannot be directly determined, and instead, only the indirect effect of the *internal* states can be observed through system outputs. For example, consider a vehicle moving in a tunnel - the rate and the velocity at which the vehicle enter and leave the tunnel can be measured directly, but the exact state inside the tunnel can only be estimated. If a system is observable, it is possible to fully reconstruct the system state from the output measurements via the state estimation process. On the other hand, in some cases, these state variables can be measured directly and be used for various statistical purposes [8]. While state estimation is one of the most common and vital steps to understand the behavior of the system and stabilizing it using control laws, the data processed during state estimation can be privacy-sensitive as explained in this thesis.

The main objective of this thesis is to investigate the privacy concerns of the state estimation process and propose solutions based on cryptography to address these privacy challenges.

1-2 Research Goals and Outline of the Thesis

The contributions of this thesis are two-fold. We address two different privacy challenges in the context of state estimation with applications to smart grids and supply chain economics. The outline and the research questions of this thesis are as follows:

- Chapter 2 is based on the following research question:

  What are the privacy concerns of smart grids in the state estimation process and how can cryptography aid in the process of designing correct, secure, verifiable and an efficient scheme that can run in a smart meter platform?

- Chapter 3 address the following research question:

  How can you preserve the privacy of a confidential dynamic supply chain model while leveraging the data (state sequences) generated by it to an external (possible) adversary for data aggregation purposes?

- Finally, in Chapter 4, we present the overall conclusions and point out future research directions for both Chapter 2 and 3.

In each chapter, we first discuss the motivation behind the problem and outline the individual contributions of that chapter. Followed by that, we discuss the technical preliminaries and mathematically formulate the problem. We then present the theoretical analysis of the proposed solution and the simulation results. Finally, we provide concluding remarks at the end of every chapter.

Chapter 2

Obfuscation-Based State Estimation in Smart Grids

In this chapter, we study the problem of protecting the user power consumption patterns and the grid topology during state estimation in smart grids based on obfuscated transformation. First, in Section 2-1, we discuss the motivation behind the problem and outline the individual contributions of this chapter. In Section 2-2, we discuss the necessary prerequisites and the type of adversarial models. In Section 2-3, we introduce the participating entities in our problem setup and state the research problem. In Section 2-4, we present the proposed solution - $Obfuscate(.)$ and explain all the sub-protocols in detail. In Section 2-5, we present the correctness, security, verification, and complexity analysis to prove that the designed protocol complies with the privacy goals of $Obfuscate(.)$. In Section 2-6, we present the simulation results to further validate the use of $Obfuscate(.)$ in practice. Finally, in Section 2-7, we summarize the conclusions of this chapter.

2-1 Introduction

2-1-1 Problem Motivation

Smart grids are widely regarded as a key ingredient to reduce the effects of growing energy consumption and emission levels [9]. By 2020, the European Union (EU) aims to replace 80% of the existing electricity meters in households with smart meters [9]. Currently, there are close to about 200 million smart meters accounting for 72% of the total European consumers [9]. This smart metering and smart grid rollout can reduce emissions in the EU by up to 9% and annual household energy consumption by similar amounts [9]. Despite the environment-friendly and the cost-cutting nature of the smart grid, deployment of smart meters at households actually raises serious data privacy and security concerns for the users. For example, with the advent of machine learning and data mining techniques, occupant activity patterns can be deduced from the power consumption measurement data [10–13]. Additionally, the configuration of the power network/grid topology can be used by attackers to launch

stealth attacks [14]. Thus, despite the apparent benefits, without convincing privacy and security guarantees, users are likely to be reluctant to take risks and might prefer conventional meters to smart meters.

State estimation in smart grids enables the utility providers and Energy Management Systems (EMS) to perform various control and planning tasks such as optimizing power flow, establishing network models, and bad measurement detection analysis. State estimation is a process of estimating the unmeasured quantities of the grid such as the phase angle from the smart meter measurement data. The operating range of the state variables determines the current status of the network which enables the operator to perform any necessary action if required. The state of the system, the network topology, and impedance parameters of the grid can be used to characterize the entire power system [15]. Traditionally, the centralized state estimation technique with the weighted-least-squares method yielded a very accurate result using the data collected from SCADA [16]. However, now due to the increased complexity and the scale of the grid size, state estimation in a wide area grid network requires multiple smart meters from different localities to share data, some of which could be hosted by a third-party cloud computing infrastructure [17] due to coupling constraints, superior computational resources, greater flexibility, and cost-effectiveness.

The problem with the current cloud computation practice is that it operates mostly over plaintexts [1,18], and hence users reveal data and computation results to the commercial cloud [18]. This poses a huge problem as the user data might contain sensitive information such as the power consumption patterns in smart meters. On top of that, there are strong financial incentives for the cloud service provider to return false results especially if the clients cannot verify or validate the results [19]. For example, the cloud service provider could simply store the previously computed result and use it as the output result for future computation problems to save computational costs. A recent breakthrough in Fully homomorphic encryption (FHE) by Craig Gentry [20] has shown the general results of secure computation outsourcing to be viable in theory. However, applying this general mechanism to compute arbitrary operations and functions is still far from practice due to its high complexity and overhead [19]. However, as shown in Section 2-4, the nature of the operations required to perform state estimation in a power grid network allows us to exploit highly efficient and verifiable obfuscation-based cryptographic solutions, thus enabling computations on the randomized data in the cloud.

Thus, motivated by the essence of regularly estimating the state variables in a power grid network for grid stabilization and reliability, we propose a privacy-aware state estimation protocol for a smart grid power network in this chapter.

2-1-2 Existing Work and Our Contributions

Numerous privacy challenges related to smart grids are pointed out in the literature in different contexts. Amongst them, the most popular and widely studied [10, 12, 21–25], is the privacy-preserving billing and data aggregation problem in smart grids. It is important to note that our main objective is different from these works since we focus on the privacy concerns of *state estimation* in smart grids. Existing literature on the smart grid state estimation problem focuses either on the problem of protecting the grid topology [2, 14, 26], or on preserving the power consumption data of the users [17, 27, 28] separately. In [14], the authors presented a new class of attacks called false data injection attacks (FDI) against state estimation in

smart grids and showed that an attacker can exploit the configuration of a power network to successfully introduce arbitrary errors into the state variables while bypassing existing techniques for bad measurement detection. The authors in [2] designed a least-budget defense strategy to protect the power system from such FDI attacks. The authors in [26] extended this problem to a non-linear state estimation and examined the possibilities of FDI attacks in an AC power network. To preserve the privacy of the user's daily activities, [17] exploits the kernel of the electric grid configuration matrix. In [27], a data obfuscation approach for an 802.11s-based mesh network was proposed to securely distribute obfuscated values along the routes available via 802.11s. Another data obfuscation approach [28] tackled this problem through advanced encryption standard (AES) scheme for the hiding the power consumption data and used elliptic-curve cryptography (ECC) for authenticating the obfuscation values that are distributed within the advanced metering infrastructure (AMI) network.

Contrary to the above work in smart grid state estimation, we focus on protecting *both* the power consumption data of the users *and* the grid topology. An open question pointed out in [17,29,30] is to provide a light-weight implementation of state estimation that can run in a smart meter platform. In this chapter, we attempt to solve this light-weight implementation issue by deploying an efficient randomized scheme where a collection of smart meters installed in a particular locality obfuscate their measurement data and send it to the lead smart meter in their respective locality. These lead smart meters, in turn, gather these randomized data and send it to the cloud service provider for performing the required computations.

The major contributions of this chapter are as follows:

- We propose $Obfuscate(.)$ - the first batch-wise state estimation scheme in smart grids with the privacy goal of protecting *both* the power consumption data of the consumers *and* the grid topology. Our scheme is based on obfuscated transformation and is proven to be efficient with no major computational overhead to the users.
- We evaluate the performance of $Obfuscate(.)$ with real-time hourly power consumption datasets of different smart meters. We use these measurement data sets under the assumption that these meters are connected to an IEEE-14 bus test grid system and a fully measured 5 bus power system. Furthermore, we evaluate the illegibility of the obfuscated datasets with respect to the original datasets.

2-2 Technical Preliminaries

2-2-1 Static State Estimation in Electric Grids

The static state estimation (SSE) in smart grids is a very well established problem with well-known techniques that rely on a set of measurement data to estimate the states at regular time intervals [31–33]. The state vector $x = [x_1, x_2, \cdots x_n]^T \in \mathbb{R}^n$ represents the phase angles at each electric branch or system node and the measurement data $z \in \mathbb{R}^m$ denotes the power readings of the smart meters. The state vector x and the measurement data z are related by a nonlinear mapping function h

$$z \;=\; h(x) \;+\; e, \tag{2-1}$$

where the sensor measurement noise e is a zero-mean Gaussian noise vector. Typically, for state estimation a linear approximation of (2-1) is used [14, 17, 34]:

$$z \;=\; \mathbf{H}\, x + e, \tag{2-2}$$

where the full column rank (m >n) measurement jacobian matrix $\mathbf{H} \in \mathbb{R}^{m \times n}$ is determined by the grid structure and line parameters [35]. The matrix $\mathbf{H}$ is known as the *grid configuration* or the *power network topology* matrix [17, 34, 35]. Typically in an electric grid $m >> n$ [36] and the best unbiased linear estimation of the state [37] is given by:

$$\hat{x} = (\mathbf{H}^T W \mathbf{H})^{-1} \mathbf{H}^T\, W\, z, \tag{2-3}$$

where $W^{-1} \in \mathbb{R}^{m \times m}$ represents the covariance matrix of the measurement noise. The covariance matrix for the measurements is taken to be a diagonal matrix $W^{-1} = \sigma^2 I$ [37]. Hence (2-3) reduces to

$$\hat{x} = (\mathbf{H}^T\, \mathbf{H})^{-1}\, \mathbf{H}^T\, z. \tag{2-4}$$

The SSE technique reduces the computational complexity for performing state estimation [38] in smart grids where the estimates are usually updated on a periodic basis. Measurement devices in current transmission systems are installed specifically catering to the needs of SSE [39]. Although the recent evolution of phasor measurement units (PMUs) are able to measure voltage and line current phasors with high accuracy and sampling rates, deployment of a large number of PMU's across the system requires significant investments since the average overall cost per PMU (including procurement, installation, and commissioning) ranges from \$40k to \$180k [40]. Hence SSE will remain an important technique to estimate the state variables at medium and low voltage levels [41]. Practically, state estimation is run only for every few minutes or only when a significant change occurs in the network [41, 42].

2-2-2 Bad Measurement Detection

Bad measurements may be introduced due to meter failures or malicious attacks and can mislead the grid control algorithms, possibly causing catastrophic consequences such as blackouts in large geographical areas. For example, a large portion of the Midwest and Northeast United States and Ontario, Canada, experienced an electric power blackout affecting a population of about 50 million [43]. The power outage cost was about \$80bn in the USA and usually, the utility operators amortize it by increasing the energy tariff, which is unfortunately transferred to consumer expenses [44]. Thus bad measurement detection or BMD is vital to ensure smooth and reliable operations in the grid.

The most common technique to detect bad measurements is to calculate the L_2-norm $\|z - \mathbf{H}\,\hat{x}\|$ where $\hat{x}$ are the estimated states. If $\|z - \mathbf{H}\,\hat{x}\| > \tau$, where τ is the threshold limit, then the measurement z is considered to be bad. This is because, intuitively, normal sensor measurements yield estimates closer to their actual values, while abnormal ones deviate the estimated values away from their true values. This inconsistency check is used to differentiate the good and the bad measurements [14]. However, this is not the case always, as exposing the grid configuration matrix $\mathbf{H}$ could make the grid vulnerable to stealth attacks [14]. Liu, Reiter, and Ning in [14] proved that a malicious entity can exploit the row and column properties of $\mathbf{H}$ when exposed, and launch *false data injection attacks* without getting

detected. The grid topology matrix **H** includes the arrangement of of loads or generators, transmission lines, transformers, and statuses of system devices [34], and is an integral part of state estimation, security, and power market design [34]. Thus, there is a strong need to protect not just the power consumption data but also the power network topology during state estimation in smart grids.

2-2-3 Cryptographic Preambles

To understand the privacy goals of our problem, we state the definitions of the following terms:

Definition 1. *(Obfuscation) [45]: Obfuscation refers to the procedure of transforming the given data in to a randomized data and performing the necessary operations on this obfuscated data. The randomized or obfuscated data can be deobfuscated by inverting the randomized transformation using the respective private keys.*

Definition 2. *(Semi-honest Adversary) [46]: A semi-honest adversary is the one who correctly follows the protocol specification with an exception that it keeps track of all the information exchanged and might possibly analyze it together with any other public information to leak sensitive data. They are also known as 'honest-but-curious' and 'passive' adversaries.*

Definition 3. *(Malicious Adversary) [46]: A malicious adversary is the one which can arbitrarily deviate from the protocol specification. Here attacks are no longer restricted to eavesdropping since the adversary might actually inject or tamper with the data provided. They are also known as 'active' adversaries.*

2-3 Problem Setup

2-3-1 Notation

Let an area $\mathcal{A}$ consist of two localities [1] or neighborhoods denoted by L_1 and L_2 as shown in Figure 2-1. The symbol S_{ij} refers to the smart meter installed at the household j situated in locality L_i. We denote by $X_i \in \mathbb{R}^{n_i \times T}$ the state sequences of all the smart meters installed in L_i for a given batch of time duration T. The electric grid configuration matrix or the power network topology of L_i is represented as $\mathbf{H}_i$ and the coupling matrices between L_i and L_j are denoted as $\mathbf{H}_{ij}$ and $\mathbf{H}_{ji}$ respectively. The symbol $[\mathbf{.}]$ denotes the obfuscation of a vector or matrix. For example, $[Z_i]$ represents the obfuscated or randomized value of the matrix $Z_i \in \mathbb{R}^{m_i \times T}$ where m_i is the number of smart meters in L_i.

2-3-2 Participating Entities

- **Utility Provider** $\mathcal{U}$: Provides utility services to the area $\mathcal{A}$ and has access to the grid configuration matrix **H**. $\mathcal{U}$ generates all the keys to initiate $Obfuscate(.)$ and

[1] For brevity, here we assume that the area consists of only two localities. The protocol presented in this chapter can easily be extended to an area consisting of more than two localities.

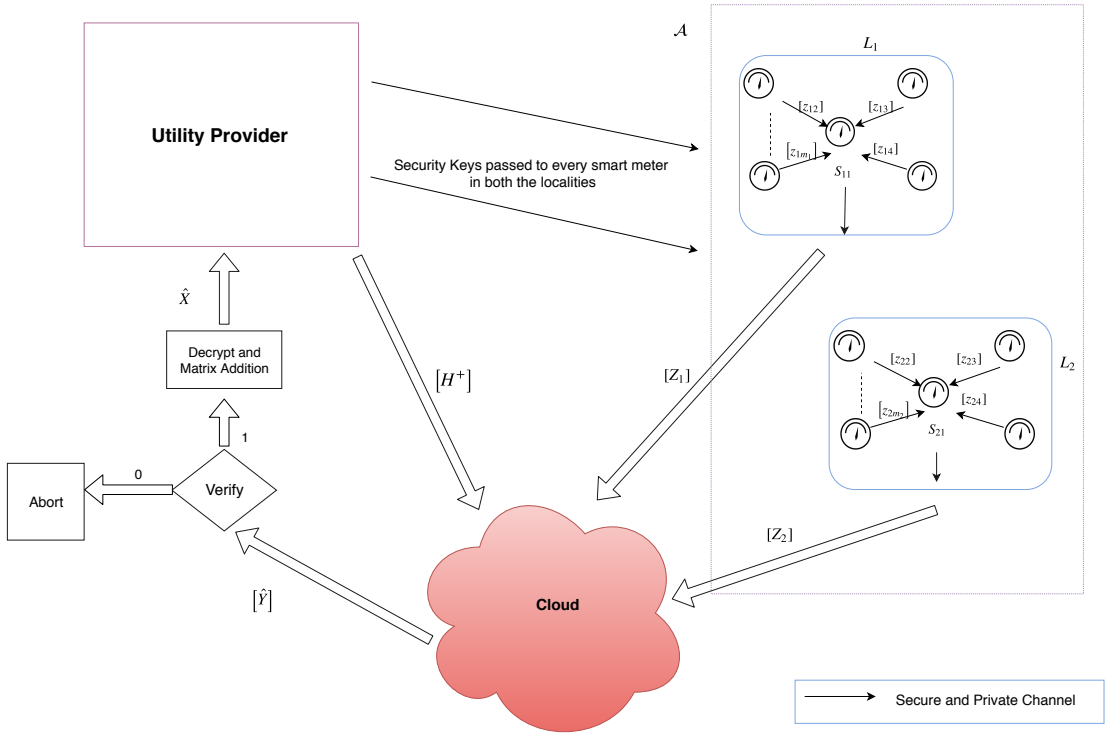


Figure 2-1: Problem setup. First, the utility provider generates a stream of random number keys for obfuscating the power consumption data and sends it to S_{ij} through a secure and private channel. All the other meters obfuscate their measurements and send it to their respective lead meter in their locality. Upon receiving the randomized data from the other meters, the lead meter obfuscates the power consumption dynamics of its locality and sends this data to the cloud. The utility provider obfuscates the grid configuration matrix and sends it to the cloud. The cloud upon receiving both the input matrices performs the necessary computations and sends the result back to the utility provider. The utility provider accepts the estimates for decision making only if the computed result passes the verification test. Otherwise, the service is simply aborted.

distributes a selected portion of these keys to the smart meters at each locality through a private channel to carry out obfuscation. $\mathcal{U}$ is a decision-making entity performing any necessary action after receiving the state variables at regular intervals.

- **Lead Smart Meter** S_{i1}: Receives the randomized data from the *other meters* connected to it and obfuscates the *dynamics* of the power consumption pattern of all the meters in its locality and then sends it to the cloud for state estimation. The lead meter at every locality is assumed to be a trusted node in the local network. A similar entity was proposed in [17] where the lead meter is connected to all the meters based on the mesh topology network. The lead meter, for instance, could be the local distributed system operator (DSO) of a particular locality.

- **Other Smart Meters** $S_{ij}\ \forall j \neq 1$: All the other meters in L_i obfuscate their measurement data and send it to the *lead meter* S_{i1} to avoid leaking information about their respective consumptions to any potential eavesdropping.

- **Cloud** $\mathcal{C}$: Computationally super efficient and hence provides computation services for $\mathcal{A}$ performing state estimation. As pointed out before, since most of the current cloud computations happen in plaintexts, modeling the cloud as a *malicious* entity is crucial in practice.

The smart meters in L_i and L_j where $j \neq i$ are considered to be *semi-honest* to each other i.e., clients living in different localities are *curious* about each other consumption data. This means that people who are situated geographically apart may try to learn information about people in other localities such as energy usage consumption pattern, pricing, etc. Also, households living in the same locality are modeled to be *honest-but-curious*. Albeit, it is natural for people living in the same locality - next to each other to have at least some prior knowledge about each other activity pattern, it is not acceptable if the neighbors can deduce the usage of a particular appliance at a given time-stamp applying techniques such as non-intrusive load monitoring [13] to the original power consumption data. Thus all the smart meters in a particular locality randomize their stream of consumption data before sending it to the lead meter.

Unlike the problem of protecting the user power consumption data from the utility provider for billing, data aggregation and other statistical purposes [12, 21–25], here we study the problem of carrying out secure state estimation by outsourcing the data to an untrusted third party. These state variables with high accuracy are essential to the utility provider for effective decision-making and providing good quality services such as demand forecasting, optimal power flow, and contingency analysis. Hence $\mathcal{U}$ here is not considered to be an adversarial entity and is *non-colluding* in nature. The utility provider's main objective is to earn the consumer trust by protecting their privacy and encouraging more user participation to install smart meters for business and commercial purposes. Investment in smart metering technology is directly impacted by customer trust in the utility operators [47]. To protect the privacy of consumers, utility providers make use of secure communication channels and databases with access control [17]. In addition, with EU's newly devised General Data Protection Regulation (GDPR) which is in effect from 25 May 2018, energy companies are liable to pay very large penalties up to €20m [48], if customer data are misused. One might argue about the need to apply a similar compliance factor to the cloud service provider, but as mentioned earlier, the

major problem and challenge specific to using cloud for computation services is that, with the current technology, most of the computations in the cloud happen in plaintext data [1, 18]. Arbitrary computations on encrypted data using FHE schemes are still under active research for effective implementation [49]. Providing data in the clear makes the cloud vulnerable to both active and passive attacks. According to the latest Microsoft security intelligence report [50], the number of attacks in the cloud environment has increased by 300% which further justifies considering the cloud as a *malicious* entity in our problem setup.

2-3-3 Research Problem and Design Goals

Problem Statement: How to protect the privacy of the power consumption data of the consumers Z *and* the grid topology matrix $\mathbf{H}$ during state estimation, while outsourcing these pieces of information to an untrusted malicious third party with the following privacy goals:

1. **Input/Output Privacy**: Neither the input data sent nor the output data computed by the cloud should be inferred by the cloud.
2. **Correctness**: Any cloud server faithfully following the $Obfuscate(.)$ protocol must be able to compute an output that can be verified successfully.
3. **Verification**: If the cloud server acts maliciously, then it should not be able to pass the utility-side verification test with a high probability.
4. **Efficiency**: The computational overhead of the clients (utility provider and the smart meters in each locality) should be minimal, as otherwise, outsourcing the computations is not practically justifiable.

Remark 1. *Nevertheless, it is important to note that local smart meters in both the localities cannot estimate the states on their own due to the coupling constraints (See* (2-5)*). The efficiency criterion is mainly considered to exploit the nearly unlimited computational resources of the cloud.*

Furthermore, since the smart meters in different neighborhoods are *semi-honest* to each other, the designed protocol should also guarantee a very low probability of inferring any sensitive information through eavesdropping and combining any other publicly available information of the localities.

2-4 $Obfuscate(.)$

Consider the problem setup depicted in Figure 2-1. The equation presented in (2-2) can be rewritten as :

$$\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} = \underbrace{\begin{bmatrix} H_1 & H_{12} \\ H_{21} & H_2 \end{bmatrix}}_{\mathbf{H}} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}, \tag{2-5}$$

where $H_1 \in \mathbb{R}^{m_1 \times n_1}$ and $H_2 \in \mathbb{R}^{m_2 \times n_2}$ are the grid configuration matrix of L_1 and L_2. The matrix $H_{12} \in \mathbb{R}^{m_1 \times n_2}$ and $H_{21} \in \mathbb{R}^{m_2 \times n_1}$ denote the coupling matrices. The measurement data and the states of locality L_i are represented by $Z_i \in \mathbb{R}^{m_i \times T}$ and $X_i \in \mathbb{R}^{n_i \times T}$ respectively. The solution to (2-5) is given by (2-4).

In general, the configuration of the power network $\mathbf{H}$ is not time-varying during the state estimation process [31–33,37], and hence the matrix $\mathbf{H}^+ = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$ can be pre-computed during the offline stage. Typically, this information is computed during the creation of the power network by the utility provider using a trusted party. Hence, the state estimation can be recast and reduced into

$$\hat{X} \;=\; \mathbf{H}^+ \, Z, \tag{2-6}$$

where $\hat{X} \in \mathbb{R}^{n \times T}$, $Z \in \mathbb{R}^{m \times T}$ and $\mathbf{H}^+ \in \mathbb{R}^{n \times m}$ with $m = m_1 + m_2$ and $n = n_1 + n_2$. Thus, the secure state estimation problem boils down to solving a matrix multiplication in (2-6) securely. The matrix $\mathbf{H}^+$ can be rewritten block-wise as follows:

$$\begin{aligned} \mathbf{H}^+ \;&=\; \left(\begin{bmatrix} H_1 & H_{12} \\ H_{21} & H_2 \end{bmatrix}^T \begin{bmatrix} H_1 & H_{12} \\ H_{21} & H_2 \end{bmatrix} \right)^{-1} \begin{bmatrix} H_1 & H_{12} \\ H_{21} & H_2 \end{bmatrix}^T, \\ &=\; \begin{bmatrix} F_1 & F_{12} \\ F_{21} & F_2 \end{bmatrix}, \end{aligned} \tag{2-7}$$

where $F_1 \in \mathbb{R}^{n_1 \times m_1}$, $F_2 \in \mathbb{R}^{n_2 \times m_2}$, $F_{12} \in \mathbb{R}^{n_1 \times m_2}$ and $F_{21} \in \mathbb{R}^{n_2 \times m_1}$. Notice from (2-6) and (2-7) that it is not possible for the lead meter in each locality to carry out the estimation process locally since the state estimate of a particular locality requires power consumption data of the other locality. Thus the lead meter collects all the obfuscated measurement data from the other meters in its locality and sends it to the cloud. The matrix $\mathbf{H}^+$ is obfuscated by the utility provider and sent to the cloud. However, it is important that the matrix $\mathbf{H}^+$ is not randomized as a whole using a single set of keys but is obfuscated block-wise with different keys for different blocks (see (2-7)). The estimation problem can be further broken down into:

$$\begin{bmatrix} \hat{X}_1 \\ \hat{X}_2 \end{bmatrix} \;=\; \begin{bmatrix} F_1 \, Z_1 + F_{12} \, Z_2 \\ F_{21} \, Z_1 + F_2 \, Z_2 \end{bmatrix}. \tag{2-8}$$

Let us denote the matrix

$$Y \;=\; \begin{bmatrix} F_1 Z_1 & F_{12} Z_2 \\ F_{21} Z_1 & F_2 Z_2 \end{bmatrix} \;=\; \begin{bmatrix} Y_1 & Y_{12} \\ Y_{21} & Y_2 \end{bmatrix}. \tag{2-9}$$

Notice from (2-8) for estimating the states, we solve the matrix multiplication of each blocks in (2-9) privately and then perform matrix addition.

The matrix multiplication or MM problem is a fundamental problem in cryptography and several solutions [51–54] had been proposed to solve it securely. However, the problem is that these algorithms were not initially designed for the cloud environment and hence, these protocols did not consider the computational asymmetry of the cloud server and the client. Moreover, these protocols use complex cryptographic protocols to encrypt the data-set (input and output), which makes them unsuitable for the computation on the cloud with large

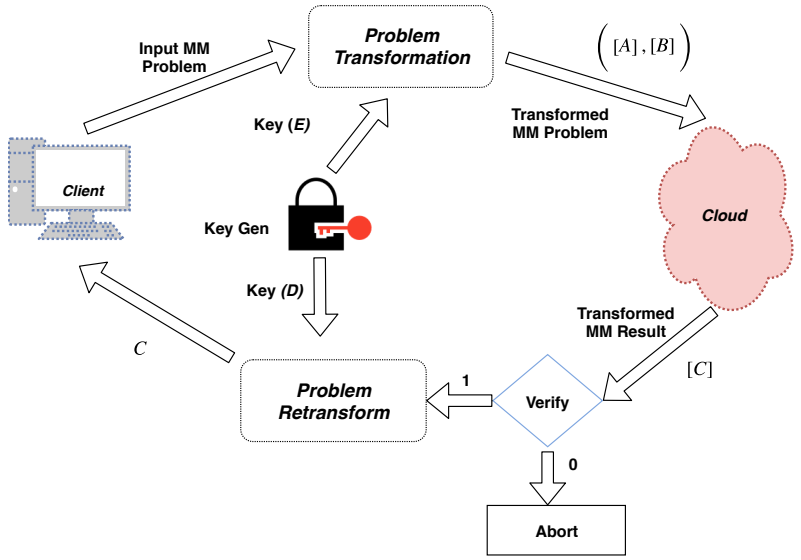


Figure 2-2: Depiction of a privacy-preserving matrix multiplication algorithm for a single client cloud setup. Client sends the input matrices A,B and decrypts the output C.

datasets due to high overhead. Furthermore, these algorithms did not consider result verification which is an essential requirement when dealing with a malicious cloud setting [55]. A secure multiparty computation (SMC) approach was also considered in [56, 57] where the computation is divided among multiple workers without allowing any participating entity to access another individual's private information. However, this approach is again not feasible to use in our problem setup since all the parties must have a comparable computing capability. Instead, we would ideally like the clients to transfer the majority of the computation load to the massive cloud servers and perform very little work computationally. Another drawback of the SMC approach is that the result verification is often proven to be troublesome and expensive since most of the times it requires expensive *zero-knowledge proofs* as a part of the verification process [58, 59].

Recently, a privacy-preserving, verifiable and efficient outsourcing algorithm for matrix multiplication to a malicious cloud was proposed in [55] utilizing linear transformation techniques.

Algorithm 1 *KeyGen*

Input λ, m_1, n_1
Generate two sets of uniform non-zero unique random numbers each of bit size λ: $N_1 \rightarrow \{\alpha_1, \alpha_2 \cdots \alpha_{n_1}\}$, $M_1 \rightarrow \{\beta_{11}, \beta_{12} \cdots \beta_{1m_1}\}$.
for $i = 1$ to n_1 **do**
    $D_1 = \alpha_i.I_{(i,i)}$
end for
for $i = 1$ to m_1 **do**
    $A_1 = \beta_{1i}.I_{(i,i)}$
end for
Output: D_1, A_1.
Repeat 1 to 10 per batch.

In this chapter, we adopt a similar approach to the one prescribed in [55] to outsource the multiplication of block matrices in (2-9) securely to the cloud. However, it is important to point out that the $Obfuscate(.)$ protocol proposed here is not a straightforward application of the protocol in [55] since the authors in [55] considered only a single client and a cloud setup as shown in Figure 2-2. The client performs the key generation, problem transformation, re-transformation and verification on his/her own. However, since we have multiple smart meters installed in different neighborhoods, the keys cannot be generated locally by the individual households because the smart meters have access only to their respective consumption data which forms only a part of the information required for state estimation. Hence, besides the key generation we also propose a key distribution scheme essential to perform $Obfuscate(.)$ meaningfully. $Obfuscate(.)$ comprises of 8 sub-algorithms in total from Algorithm 1 - 8.

First we present the $KeyGen()$ given by Algorithm 1:

- $KeyGen(1^{\lambda}, m_1, n_1)$: The $KeyGen()$ algorithm takes in the input security parameter λ and generates a total of $n_1 + m_1$ *non-zero* random numbers each of bit size λ. These random numbers are in turn used to generate the key matrices of size $\mathbb{R}^{m_1}$ and $\mathbb{R}^{n_1}$.

Every time the utility provider $\mathcal{U}$ invokes the $KeyGen()$ algorithm, different set of *non-zero* random numbers of bit size λ are generated. The complete key matrices generated by $\mathcal{U}$ for a given batch of duration T is given by Table 2-1. Once $\mathcal{U}$ generates all the key matrices,

Table 2-1: Key generation protocol run by the utility provider per batch

Protocol	Output
$KeyGen(1^{\lambda}, n_1, m_1)$	$D_1,\ A_1$
$KeyGen(1^{\lambda}, n_2, m_2)$	$D_5,\ A_2$
$KeyGen(1^{\lambda}, n_2, A_1)$	$D_3,\ A_1$
$KeyGen(1^{\lambda}, n_1, A_2)$	$D_6,\ A_2$
$KeyGen(1^{\lambda}, T)$	D_2
$KeyGen(1^{\lambda}, T)$	D_4

a selected portion of these keys are distributed by $\mathcal{U}$ as shown in Figure 2-3 to the smart meters. The $KeyDist()$ protocol for L_i is given by Algorithm 2.

Algorithm 2 $KeyDist$

1: **Input**: A_i, D_{2i}
2: **Set** $a_{ij} = \frac{1}{\beta_{ij}}$
3: **for** $j = 2$ to m_i **do**
4: Send a_{ij} to S_{ij} through private channel $\{i, j\}$
5: **end for**
6: Send a_{i1} and D_{2i} to S_{i1} through private channel $\{i, 1\}$
7: **Repeat** 1 to 6 per batch.

After the $KeyDist()$ algorithm, the matrix transformation $\psi_K()$ is carried out by the respective entities using their respective keys K.

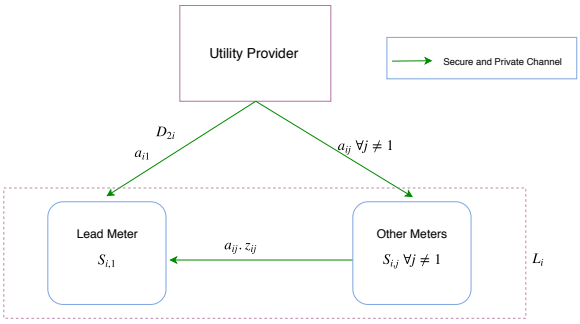


Figure 2-3: A triangular key distribution scheme. In every neighborhood L_i, the other smart meters receive a random number key a_{ij} of bit size λ per batch for obfuscating their consumption data z_{ij}. The lead meter also receives a key a_{i1} for randomizing its own measurement data and a matrix D_{2i} for obfuscating Z_i which is then sent to the cloud.

- Matrix Transformation $\psi_K()$: For every new input matrix, $\psi_K()$ invokes and transforms the input in order to preserve the privacy. This operation dominates the client-side computation cost, but is still significantly less compared to the computations performed by the cloud.

The matrix transformation for a given input matrix F_1 and Z_1 are given by Algorithm 3 and 4 respectively. Table 2-2 summarizes the complete matrix transformation protocol.

Algorithm 3 MatrixTrans $\psi_K(F_1)$

Input D_1, A_1
for $i = 1$ to n_1 **do**
    $[F_1(i,:)] = D_1(i,i).F_1(i,:)$
end for
for $i = 1$ to m_1 **do**
    $[F_1(:,i)] = F_1(:,i).A_1(i,i)$
end for
Output $[F_1]$

Algorithm 4 *MatrixTrans* $\psi_K(Z_1)$

Input D_2
for $j = 2$ to m_1 **do**
    Send $z'_{1j} = a_{1j}.z_{1j}$ to S_{11}.
end for
S_{11} constructs $Z'_1 = A_1^{-1}.Z_1$. ▷ $a_{1j} = 1/\beta_{1j}$
for $i = 1$ to T **do**
    $[Z_1(:,i)] = Z'_1(:,i).D_2(i,i)$
end for
Output $[Z_1]$

Next, the obfuscated matrix $\mathbf{H}^+$ and the measurement data matrix Z_i are sent by $\mathcal{U}$ and S_{i1} respectively to the cloud $\mathcal{C}$ to perform the $Compute_\psi()$ algorithm given in Algorithm 5.

Table 2-2: Matrix transformation protocol run per batch

Protocol	Keys	Run by	Output
$\psi_K(F_1)$	D_1, A_1	$\mathcal{U}$	$[F_1]$
$\psi_K(F_2)$	D_2, A_2	$\mathcal{U}$	$[F_2]$
$\psi_K(F_{12})$	D_6, A_2	$\mathcal{U}$	$[F_{12}]$
$\psi_K(F_{21})$	D_3, A_1	$\mathcal{U}$	$[F_{21}]$
$\psi_K(Z_1)$	D_2, A_1^{-1}	S_{11}	$[Z_1]$
$\psi_K(Z_2)$	D_4, A_2^{-1},	S_{21}	$[Z_2]$

- $Compute_\psi([F_1], [Z_1])$: This sub-algorithm performs the computation on the cloud server. It computes MM as $\psi([F_1], [Z_1]) = (D_1 F_1 A_1).(A_1^{-1} Z_1 D_2)$

Algorithm 5 $Compute_\psi$

1: **Input** $[F_1]$, $[Z_1]$
2: $\mathcal{C}$ computes $[Y_1] = [F_1] . [Z_1]$
3: **Output** $[Y_1]$

Table 2-3 presents the $Compute_\psi()$ protocol run by the cloud server for estimating the state samples. Upon computing the Y matrix, the cloud sends the computed result to the utility

Table 2-3: Computation protocol run by the cloud server $\mathcal{C}$ per batch

Protocol	Output
$Compute_\psi([F_1], [Z_1])$	$[Y_1]$
$Compute_\psi([F_2], [Z_2])$	$[Y_2]$
$Compute_\psi([F_{21}], [Z_1])$	$[Y_{21}]$
$Compute_\psi([F_{12}], [Z_2])$	$[Y_{12}]$

provider $\mathcal{U}$ to execute the verification step.

- $Verify([Y], \gamma)$: This sub-algorithm computes

$$Q = ([F].([Z].\gamma)) - ([Y].\gamma).$$

where γ is a binary key matrix of size T i.e. $\gamma \in \{1, 0\}^T$. We introduce the binary column matrix key γ to minimize the complexity of computation, since the matrix-vector multiplication only cost quadratic time. The verification protocol for L_i is given by Algorithm 6.

The verification step serves as the BMD test in our setup and is run for all the four block matrices given by (2-9). Table 2-4 presents the verification protocol. The results are accepted

Algorithm 6 $Verify([Y_i], \gamma_i)$

1: **Input**: $[Y_i]$, $[F_i]$, $[Z_i]$
2: $\mathcal{U}$ generates $\gamma_i \in \{0,1\}^T$ and sends it to S_{i1} through a private channel.
3: S_{i1} computes $Z_{\gamma_i} = [Z_i].\gamma_i$ and sends it back to $\mathcal{U}$.
4: $\mathcal{U}$ computes $Q_i = [Y_i].\gamma - F_i.Z_{\gamma_i}$
5: **if** $(Q_i == \{0, 0, \cdots, 0\}^T)$ **then**
6: return (1)
7: **else**
8: return (0)
9: **end if**

Table 2-4: Verification protocol run by the utility provider per batch

Protocol	Output
$Verify([Y_1], \gamma_1)$	Q_1
$Verify([Y_2], \gamma_2)$	Q_2
$Verify([Y_{12}], \gamma_2)$	Q_{12}
$Verify([Y_{21}], \gamma_1)$	Q_{21}

only if the cloud server passes all the four verification tests. If the verification is positive, then it means that no false data has been injected into the datasets by the cloud. This suffices to conclude that no bad measurements are introduced in the network. After the verification test, $\mathcal{U}$ runs the $Decrypt()$ algorithm given in Algorithm 7.

- $Decrypt(Y, K)$: This protocol decrypts the matrix Y using its respective key K.

Table 2-5 summarizes the decryption protocol carried out for all the four block matrices. Once, all the four blocks of the Y matrix are decrypted, $\mathcal{U}$ carries out the protocol given in Algorithm 8 to finally arrive at the state estimates.

Algorithm 7 $Decrypt([Y_1], K)$

1: **Input** $[Y_1]$ and the respective keys D_1 and D_2
2: Compute D_1^{-1} and D_2^{-1} ▷ Since the key matrices are diagonal square invertible matrices, inversion only cost linear time computation.
3: Compute $Y_1 = D_1^{-1}.[Y_1].D_2^{-1}$
4: **Output** Result Y_1

Algorithm 8 $MatrixAdd(Y)$

1: **Input** Y_1, Y_2, Y_{12}, Y_{21}
2: Compute $\hat{X}_1 = Y_1 + Y_{12}$
3: Compute $\hat{X}_2 = Y_{21} + Y_2$
4: **Output** Result $\hat{X}_1$, $\hat{X}_2$

Table 2-5: Decryption protocol run by the utility provider

Protocol	Keys (K)	Output
$Decrypt([Y_1], K)$	D_1, D_2	Y_1
$Decrypt([Y_2], K)$	D_5, D_4	Y_2
$Decrypt([Y_{21}], K)$	D_3, D_2	Y_{21}
$Decrypt([Y_{12}], K)$	D_6, D_4	Y_{12}

2-5 $Obfuscate(.)$ - Theoretical Analysis

In this section, we show that the $Obfuscate(.)$ proposed in Section 2-4 is able to comply with all the privacy design goals stated in Section 2-3-3: correctness, security, verifiability, and efficiency.

2-5-1 Correctness Analysis

If the smart meters, utility provider, and the cloud follow $Obfuscate(.)$ as per the instruction, then $Obfuscate(.)$ produces correct results for all the four matrix multiplications. This follows from a simple proof:

Proof. The utility provider $\mathcal{U}$ first transforms the matrix F_1 into $[F_1] = D_1 F_1 A_1$ and the lead smart meter in L_1 transforms the matrix $Z_1' = A^{-1} Z_1$ into $[Z_1] = A_1^{-1} Z_1 D_2$. The cloud server computes

$$[Y_1] = [F_1] \,.\, [Z_1] = (D_1 F_1 A_1) \,.\, (A_1^{-1} Z_1 D_2) = D_1 Y_1 D_2.$$

Then in the problem re-transformation step given by the decryption algorithm, $\mathcal{U}$ computes Y_1 where,

$$\begin{aligned} Y_1 &= D_1^{-1} [Y_1] D_2^{-1}, \\ Y_1 &= F_1 . Z_1. \end{aligned} \tag{2-10}$$

□

The above analysis holds for all the $Compute_{\psi}(.)$ presented in Table 2-3, thereby proving the correctness of $Obfuscate(.)$.

2-5-2 Security Analysis

1. Input Privacy: Since the cloud server has access only to the transformed randomized input matrices $[F]$ and $[Z]$, it cannot not retrieve the original input matrices F and Z. Furthermore, the security keys in Table 2-1 do not leak any information about the original input matrices. This can be seen from the following proof:

   Proof. The key matrix A_1 and A_2 are diagonal matrices with each element being a random real number of λ bit long. There are $2^{m_i \lambda}$ possibilities of A_i matrix where

$i \in \{1, 2\}$. For diagonal matrices D_1 and D_2, there are in total $2^{n_1\lambda+T\lambda}$ possibilities. Thus for a single block F_1 in matrix Y, there are a total of $2^{(m_1+n_1+T)\lambda}$ possible choices of key matrices, which is an exponential bound quantity in terms of (m_1, n_1, T). Thus the cloud does not recover any meaningful information. □

2. Output Privacy: Similarly to the input privacy analysis, the output result is also protected. The resulting obfuscated matrix does not leak any information to the cloud, even if the cloud records all the computed results. Besides, for every batch, the utility provider generates new security keys given in Table 2-1. Thus our scheme is similar to that of one-time-pad encryption system thereby making it resistant to known-plain-text attack (KPA) or chosen-plain-text-attack (CPA) [55].

2-5-3 Verification Analysis

Since in a malicious threat model, the cloud server may deviate from the actual instructions of the given protocol, we equip $Obfuscate(.)$ with a result verification algorithm to validate and verify the correctness of the result. The proof that a wrong or an invalid result never passes the verification step follows from the *total probability theorem* as followed in [55, 60].

Proof. If the cloud produces the correct result Y_1 then

$$Q_1 \;=\; (\; [F_1]\,.\,[Z_1] - [Y_!]\;) \;=\; [0, 0, \cdots 0]^T. \tag{2-11}$$

If the cloud produces the wrong result, then

$$Q_1.\gamma_1 \neq [F_1]\,[Z_1]\,.\gamma - [Y_1]\,.\gamma,$$

i.e. there exists atleast a row in Q_1 which is not equal to zero.

$$Q_1\gamma_1 = [q_1, \cdots q_{m_1}]^T.$$

Let the row $q_i \neq 0$ where

$$q_i = \sum_{j=1}^{T} Q_{1i,j}.\gamma_j = Q_{1i,1}.\gamma_1 + \cdots Q_{1i,k}.\gamma_k + Q_{1i,T}.\gamma_T. \tag{2-12}$$

There exists at least one element in this row which is not equal to zero. Let $Q_{1i,k} \neq 0$

$$q_i = Q_{1i,k}.\gamma_k + \Gamma \;\;,$$

where $\Gamma = \sum_{j=1}^{T} Q_{1i,j}.\gamma_j - Q_{1i,k}.\gamma_k$. Applying the total probability theorem yields,

$$\Pr(q_i = 0) = \Pr[(q_i = 0)|(\Gamma = 0)]\Pr[\Gamma = 0] + \Pr[(q_i = 0)|(\Gamma \neq 0)]\Pr[\Gamma \neq 0], \tag{2-13}$$

$$\begin{aligned} \Pr[(q_i = 0)|(\Gamma = 0)] = \Pr[\gamma_k = 0] \quad &= 1/2 \;\;, \\ \Pr[(q_i = 0)|(\Gamma \neq 0)] \leq \Pr[\gamma_k = 1] \quad &= 1/2 \;\;. \end{aligned} \tag{2-14}$$

Substituting (2-14) in (2-13), we arrive at

$$\begin{aligned} &\Pr[(q_i = 0)] \leq 1/2\Pr[\Gamma = 0] + 1/2\Pr[\Gamma \neq 0], \\ &\Pr[(q_i = 0)] \leq 1/2(1 - \Pr[\Gamma \neq 0]) + 1/2\Pr[\Gamma \neq 0]. \end{aligned} \tag{2-15}$$

$$\Pr[(q_i = 0)] \leq 1/2. \tag{2-16}$$

If the verification process is run p times, then

$$\Pr[(q_i = 0)] \leq 1/2^p. \tag{2-17}$$

□

The value p reveals the trade-off between computational efficiency and verifiability. Theoretically $p \geq 80$ is sufficient [55] to ensure *negligible* probability for the cloud to pass the verification test despite producing wrong result. However, here we take $p = 20$, as in practice is acceptable [55, 60] with $1/2^{20} \approx 1$ million. The verification process fails to detect a wrong result one in a million times.

2-5-4 Efficiency Analysis

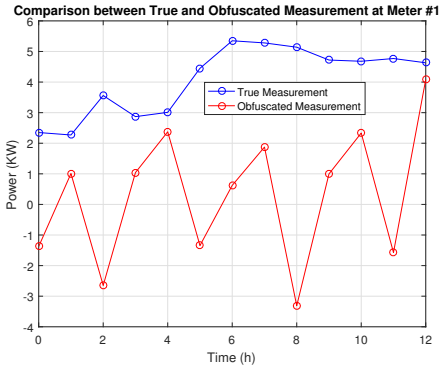
Table 2-6: Computation complexity analysis of $Obfuscate(.)$

Client Side Computations			Cloud Computations
Utility Provider $\mathcal{U}$	Lead Meter S_{i1}	Other Meters S_{ij}	
$KeyGen$ - $O(m+n+T)$	–	–	$Compute_{\psi}([F_1],[Z_1])$ - $O(n_1 m_1 T)$
MatrixTrans $\psi_K(.)$ - $O(n_1 m_1 + n_1 m_2 + n_2 m_1 + n_2 m_2) = O(nm)$	$O(m_i T)$	O(1)	$Compute_{\psi}([F_{12}],[Z_2])$ - $O(n_1 m_2 T)$
$Verify$ - $O(n_1 T + n_2 T) = O(nT)$	$O(m_i T)$	–	$Compute_{\psi}([F_{12}],[Z_2])$ - $O(n_2 m_1 T)$
$Decrypt$ - $O(n_1 T + n_2 T) = O(nT)$	–	–	$Compute_{\psi}([F_2],[Z_2])$ - $O(n_2 m_2 T)$
$MatrixAdd$ - $O(n_1 T + n_2 T) = O(nT)$	–	–	$Compute_{\psi}([\mathbf{H}^+],[Z])$ -,$O(nmT)$
Total Client-Side Computation Cost $\approx O(nm + mT + nT)$			Total Cloud Computation Cost = $O(nmT)$

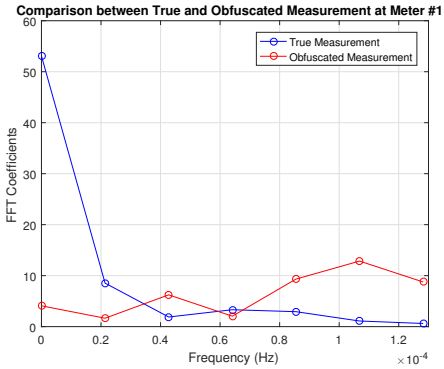
In this section, we carry out the computation complexity analysis to prove the efficiency of $Obfuscate(.)$. The computational cost of each step in $Obfuscate(.)$ has been analyzed and is given in Table 2-6. Besides, the $KeyDist()$ protocol introduces a communication cost of $O(m)$ since $\mathcal{U}$ distributes the key a_{ij} to all the smart meters through a private channel for obfuscating their measurement data. From Table 2-6, it is clear that the computations performed by the client side are substantially lower than that of the cloud server. Due to the diagonal structure of the key matrices, the problem transformation step given by Algorithm 3 and 4 only costs $O(nm+mT)$. The asymptotic complexity [55] of the client side computation is only $O(nm+mT+nT)$, and thus, outsourcing the computation yields a performance gain of $O(\frac{1}{n} + \frac{1}{m} + \frac{1}{T})$. Clearly, when the number of state variables n, the number of smart meters m and the time duration of the batch T increases, the clients will achieve a higher performance gain. Especially, by the year 2020, with the increase in the number of smart meters m as the EU is aiming to replace 80% of electricity meters with smart meters [9], $Obfuscate(.)$ will significantly reduce the computational overhead of its clients in the long run.

2-6 Simulation Results

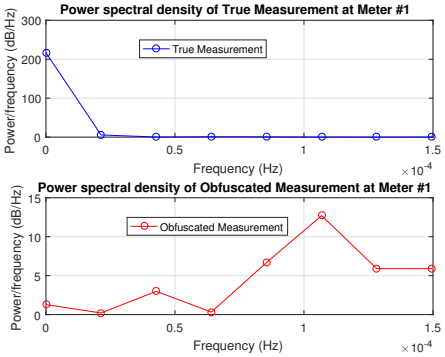
In this section, we evaluate the degree of the obscurity of $Obfuscate(.)$ using two case study examples: a fully measured 5-bus system and the IEEE 14-bus system with *real-time* power consumption data. We start with a fully measured 5-bus system and the structure of the $\mathbf{H}$ matrix for this system can be found in the Appendix 4-2 Section -1. In this case, the total



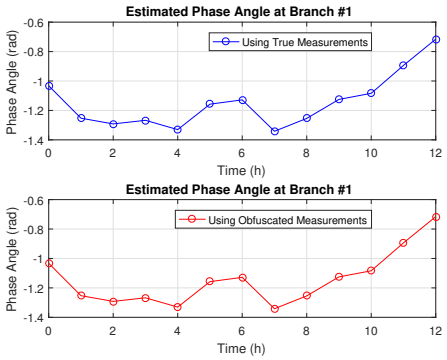
(a) Original and obfuscated time domain data from meter #1



(b) Original and obfuscated frequency domain data from meter #1



(c) Power spectral density of true and obfuscated measurement data



(d) Estimate state value at branch #1

Figure 2-4: Illustration of the efficacy of $Obfuscate(.)$ in a fully measured 5-bus power system. (a) shows the power consumption data of true and obfuscated measurement in time domain. (b) shows the power consumption data of true and obfuscated measurement in frequency domain. (c) shows the power spectral density of the original and obfuscated datasets. (d) shows that the estimated state from the obfuscated value (bottom) is same as the estimated state from the original data (top).

number of meters $m = 10$ and the state variables $n = 4$. We consider $m_1 = 4$, $m_2 = 6$ and $n_1 = n_2 = 2$ and the duration of every batch to be 13 hours. Note in practice, smart meters can sample at much higher frequencies [61]. Research on disaggregating electricity load has been conducted on smart meter readings with a fine granularity of frequency between 1 Hz to 1 MHz [61]. The authors in [17] collected real-time power consumption data of both residential and office spaces with a sampling rate of 1 Hz and, hence in practice the number of data points collected per batch T could be in order of tens of thousands. However, due to the unavailability of such *high-frequency* measurement data, we restrict the size of T. Since, we had access to only *hourly* power consumption data we restrict $T = 13$. Even though the size of the matrix $Z \in \mathbb{R}^{m \times T}$ is small, the state estimation still cannot be performed locally due to the *coupling* constraints between the two localities. Upon inspecting the power consumption values of all the meters, we found these values are mostly 4 to 5 decimal digits long. Hence, to obfuscate this measurement data, we use a key size of length $\lambda = log_2(10^5) \approx 16$ bits.

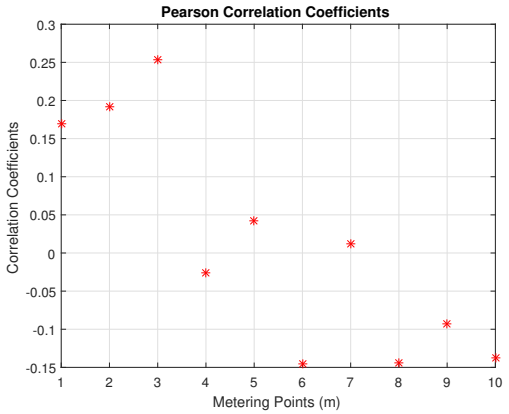
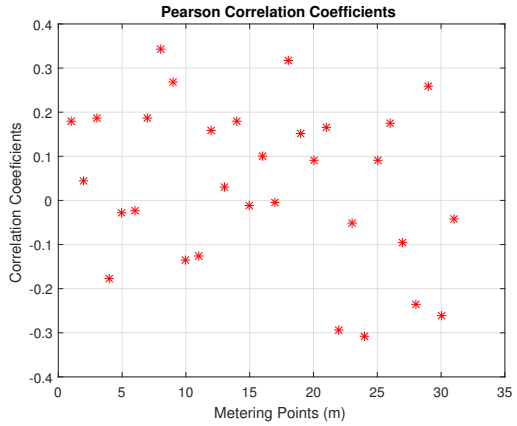
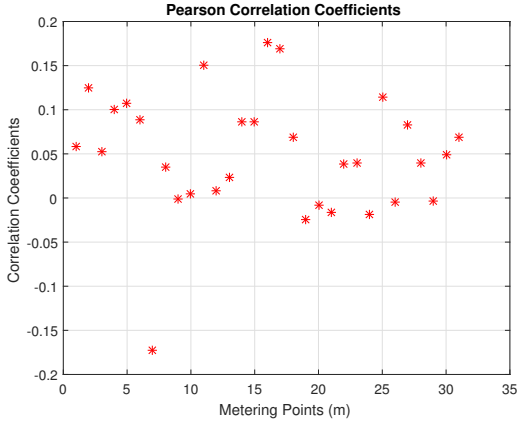


Figure 2-5: Pearson correlation coefficients of all the metering points in a fully measured 5-bus power system

Figure 2-4 shows the illegibility of the $Obfuscate(.)$ for a fully measured 5-bus power system. Illegibility [17] measures the level of difficulty of interpreting and mining data to the malicious cloud server. From Figure 2-4(a), we can see the original power consumption data of a household (blue) is always positive, whereas, the obfuscated data (red) show negative power readings and behave more as random variables. The degree of obscurity becomes more clear when transforming these datasets into the frequency domain. Figure 2-4(b) shows that the original data consists mostly of low-frequency components, whereas the obfuscated data exhibits high-frequency components. This can also be inferred from the power spectral density plot shown in Figure 2-4(c). Clearly, we can see that the original data (top) consists of a higher power in low-frequency regions, whereas the obfuscated dataset (bottom) behaves exactly the opposite consisting of a higher power in high-frequency regions. Nevertheless, as it can be seen from Figure 2-4(d), the estimated states from these obfuscated datasets are exactly the same as that of the original measurement data. Thus, $Obfuscate(.)$ does not degrade the quality of the estimate of the state variables. Furthermore, to evaluate the resilience of $Obfuscate(.)$, we estimate the *Pearson's correlation coefficient*. The Pearson's correlation coefficient gives us the measure of the degree of similarity between two signals. The correlation coefficient between two identical signals in phase is always 1 while two identical signals out of phase (phase difference = 180°) is -1. Figure 2-5 depicts the plot showing the Pearson



(a) $T = 13$

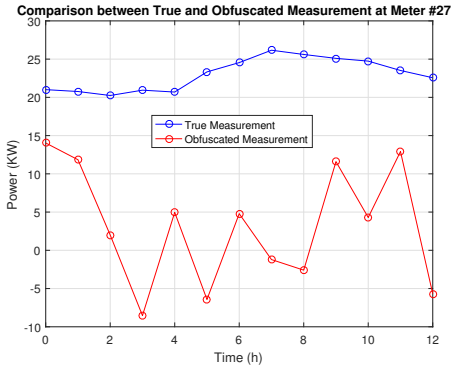


(b) $T = 360$

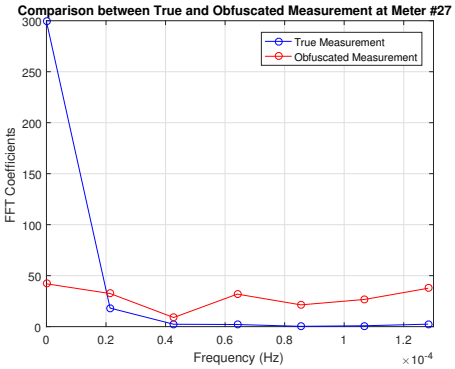
Figure 2-6: Pearson correlation coefficients of all the metering points in an IEEE 14 bus system.

correlation coefficient of all the metering points of the 5-bus systems. It can be seen that the correlation between the original and the obfuscated datasets are mostly smaller than 0.2 for all the metering points. This implies that it is very hard for any pattern recognition and data mining algorithm to infer information about the original power consumption pattern of the smart meters from the obfuscated datasets [17].

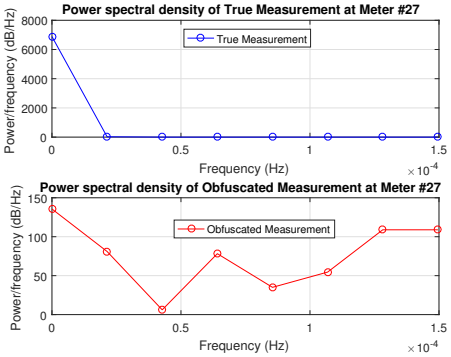
Next, we evaluate the degree of obscurity for an IEEE 14 bus system. The $\mathbf{H}$ matrix for the 14 bus system is extracted from MATPOWER, a power system simulation tool [62]. In this case, the number of metering points $m = 31$ and the number of state variables $n = 13$. We further partition the number of meters and state variables for L_1 and L_2 as $m_1 = 15$, $m_2 = 16$ and $n_1 = 6$, $n_2 = 7$. Figure 2-7 depicts the time domain, frequency domain data and the estimated states from the original and obfuscated measurement data. Comparing Figure 2-7 and 2-4, we arrive at similar conclusions for a 14-bus system to that of a 5-bus system. Figure 2-6(a) shows the correlation coefficients of all the 31 metering points for $T = 13$ and it can be seen that the value is mostly less than 0.3. Note from Figure 2-6(b) that as expected when the number of measurement data samples is increased i.e., when the value of T was increased from 13 to 360, the correlation coefficient was found to be less than 0.15 which



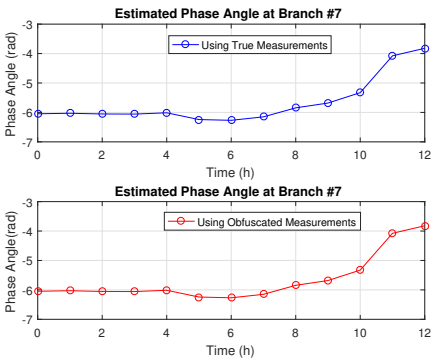
(a) Original and obfuscated time domain data from meter #27



(b) Original and obfuscated frequency domain data from meter #27



(c) Power spectral density of true and obfuscated measurement data



(d) Estimate state value at branch #7

Figure 2-7: Illustration of the efficacy of $Obfuscate(.)$ in an IEEE 14-bus power system. (a) shows the power consumption data of true and obfuscated measurement in time domain. (b) shows the power consumption data of true and obfuscated measurement in frequency domain. (c) shows the power spectral density of the original and obfuscated datasets. (d) shows that the estimated state from the obfuscated value (bottom) is the same as the estimated state from the original data (top).

makes this scheme practically secure for estimation with fine granular high-frequency meter readings. Also, in this case, since each key size is 16 bits, a semi-honest neighbor (other smart meters) trying to infer the power consumption of a household in the same locality has about $2^{16} \approx 66k$ possibilities for every batch. Naturally, when the time duration *per batch* drops down to every few minutes with high-frequency datasets, the task becomes even more cumbersome for a semi-honest adversary to deduce the appliance usage patterns of his/her neighbor living in the same locality.

However, $Obfuscate(.)$ still has a shortcoming since it cannot preserve the privacy of zero elements. The power grid topology matrix $\mathbf{H}$ is, in general, a full column rank and a *sparse* matrix. However, the matrix $\mathbf{H}^+$ is less sparse than $\mathbf{H}$ and is likely to be dense. Upon inspecting the *sparsity* pattern of the $\mathbf{H}^+$ matrix for both the 5-bus and 14-bus power system, we found that the $\mathbf{H}^+$ matrix for the 14-bus was about 20% sparse, whereas the $\mathbf{H}^+$ matrix for the 5-bus power system was completely dense. Exposing the sparsity pattern of the $\mathbf{H}^+$ matrix to the cloud may, in turn, reveal some information about the structure of the $\mathbf{H}$ matrix which is undesirable. Thus to confront such sparse attacks, we introduce the matrix

$$\mathbf{H}^+_\Delta = \mathbf{H}^+ + \Delta \ ,$$

where the matrix Δ is 100% dense. The utility provider $\mathcal{U}$ sends the matrix $\mathbf{H}^+_\Delta$ instead of $\mathbf{H}^+$ to the cloud which computes

$$X_\Delta = (\mathbf{H}^+ + \Delta)\, Z \ .$$

$\mathcal{U}$ then computes the product $\Delta\, Z$ by invoking $Obfuscate(.)$ again. Later, the original state estimates can be retrieved by $\mathcal{U}$ as

$$\hat{X} = X_\Delta \ - \ \Delta\, Z \ .$$

Note that this step does not incur any major computational overhead on the utility provider since it requires another simple invocation of $Obfuscate(.)$.

2-7 Conclusions

In this chapter, we proposed a batch-wise state estimation problem in power networks with the objective of protecting *both* the grid configuration *and* power consumption data of the smart meters. We formulated a weighted-least-squares problem and reduced the state estimation problem into a matrix multiplication problem of four block matrices. This allowed us to exploit highly efficient and verifiable obfuscation-based cryptographic protocols. We designed $Obfuscate(.)$ - a randomization scheme that supports error-free estimation between the original and obfuscated datasets without compromising on the accuracy of the state variables essential to the utility provider, and is proven to be correct, secure, and verifiable. Computation complexity analysis shows the efficiency and the practical applicability of $Obfuscate(.)$. We further evaluated the performance of $Obfuscate(.)$ in terms of its illegibility and resilience with a real-time hourly power consumption data in an IEEE 14 bus and a fully measured 5 bus power system. Our simulations reveal a high degree of obscurity making it hard for the malicious cloud server to infer any information regarding the behavioral pattern of the consumers and the network topology from the obfuscated datasets. Furthermore, we also discussed the problem of revealing the sparsity structure of the pseudo-inverse of the grid topology matrix and proposed a solution to resist such sparse attacks.

Chapter 3

Protecting the System Model through Differential Privacy

In this chapter, we study the problem of protecting the system model while leveraging the state sequences generated by that model for data aggregation purposes using differential privacy. In Section 3-1 we discuss the motivation behind the problem and outline the contributions of this chapter. In Section 3-2, we present the necessary prerequisites on differential privacy. Followed by that, we present the problem formulation along with the required definitions and state the research problem in Section 3-3. In Section 3-4, we propose a noise adding mechanism based on differential privacy and derive an analytical expression to estimate the minimum noise level required to ensure DP. We then present the simulation results and study the effect of the proposed DP-mechanism w.r.t the various privacy design parameters and utility in Section 3-5. Finally, in Section 3-6, we summarize the conclusions.

3-1 Introduction

3-1-1 Problem Motivation

Innovative business models are the key to success in any industry [63]. Companies make significant capital investments to develop innovative models for improving the performance of their existing systems [64]. Many companies restrict the use of their ideas by filing patents or by hiding certain features of their model and capitalize on it to generate revenues and profits for their business [65]. Thus, protecting a model as a confidential trade secret play an important role in the growth and innovation of a company. For example, beverage companies successfully ensure that the syrup formula cannot be reverse engineered using their beverage available in the market. However, when dealing with dynamical systems, the problem of protecting the model becomes challenging since system identification techniques can be employed for identifying a black-box system model or to extract the parameters of a grey-box system. Furthermore, with the advent of machine learning, data mining techniques can be applied for

system identification [66, 67]. Such techniques help competitors unravel the trade secret of the target company. This advocate for methods that introduce ambiguity and uncertainty in the mind of an external adversary wanting to use information associated with the model for statistical purposes.

One of the most important quantities to be computed in surveys and audits is the aggregate information. Data aggregation enables performing data mining applications for understanding important phenomena, such as traffic congestion patterns, influenza outbreaks, etc. [68]. For example, consider a case where an external party wants to periodically know the number of products produced and sold in a particular time period, and use it to compute the average for statistical analysis. Companies sometimes voluntarily release this information in the form of their production report, sales report etc, and in some cases, they are mandated to release even the amount of raw materials used in the production of their product. For example, in the case of beverage industries, the main raw material is water, and the amount of water used in production must be disclosed to a governmental body for keeping a check on the underground water levels. However, as illustrated in the motivating example below, periodically releasing this information may also be used to identify sensitive model parameters. Thus, there is a fundamental need to preserve the system model as well as share the information generated by the model with a reasonable utility.

Motivating Example from Supply Chain Economics

Consider an example of supply chain economics depicted in Figure 3-1 which involves three different parties: Supplier (S), Producer (P) and Retailer (R). S purchases the quantity $u(k)$ of raw materials at each day k and discards a fraction δ_1 of raw materials when shipping a fraction α_1 to P. P transforms these raw materials into finished products and sells a fraction of α_2 to R while discarding a fraction of δ_2 due to faults, low quality etc. Finally, R returns a fraction β_3 of defective products every day and sells a fraction of γ_3 to customers.

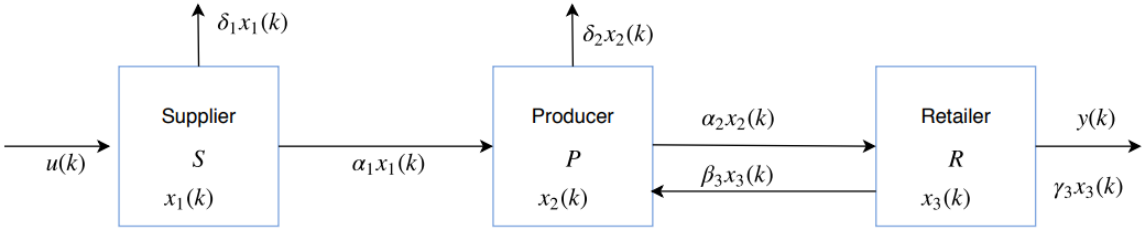


Figure 3-1: A simple supply chain model showing the internal flow of information between the supplier, producer and retailer.

This supply chain model can be recast into a discrete-time linear state space equation as follows:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 1-\alpha_1-\delta_1 & 0 & 0 \\ \alpha_1 & 1-\alpha_2-\delta_2 & \beta_3 \\ 0 & \alpha_2 & 1-\beta_3-\gamma_3 \end{bmatrix}}_{A} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u(k). \tag{3-1}$$

The state $x_1(k)$ represents the amount of raw material in S, $x_2(k)$ and $x_3(k)$ denotes the products in P and R respectively. The output $y(k)$ represents the products sold to customers. The system model matrix A contains information about the percentage of products discarded in each chain, the percentage of defective products returned by the retailer to the producer. Typically companies would like to keep such internal information private when disclosing the state vector $x(k)$ to an external organization for survey and auditing purposes. Exposing the system matrix might damage the reputation of each party in the supply chain and may even result in the breach of trust among the customers. For example, consider exposing the information δ_1 and β_3 to the public. A higher δ_3 implies that a large percentage of raw material supplies have been discarded due to poor quality. Higher β_3 implies that the percentage of defective products produced is high. Information such as the percentage of products discarded in each chain or percentage of defective products may also give insight into sensitive information such as the quality and efficiency of the production machine, thinking pattern behind rejecting products etc. Furthermore, if a competitor gets hold of the supply chain model i.e. the A matrix of the target company, then it could very likely predict the future production of the target company with high accuracy and beat them to the market. Thus, the system matrix A must be protected while releasing the information of the state vectors for data aggregation purposes.

Let us now discuss the scenario of releasing the state samples after a certain time duration of T days without perturbing the state samples, i.e., no privacy. These state vectors are typically required in surveys and audits for measuring the aggregate amount of products from the supplier, producer and retailer side for a period of time. Consider the input $u(k)$ to be a *Dirac delta function*[1] of magnitude C i.e. $u(k) = C\delta$. This means once, for every T days, the supplier S purchases a raw material of quantity C. Thus for $k \in (0, T]$, the state space equation in (3-1) reduces to

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 1-\alpha_1-\delta_1 & 0 & 0 \\ \alpha_1 & 1-\alpha_2-\delta_2 & \beta_3 \\ 0 & \alpha_2 & 1-\beta_3-\gamma_3 \end{bmatrix}}_{A} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix}. \tag{3-2}$$

Let the external query at a given time index k be

$$Q_k = x_k,$$

Upon repeating this query for a period T, the adversary arrives at the following relation

$$\underbrace{\begin{bmatrix} x_1 & x_2 & \cdots x_T \end{bmatrix}}_{\mathbf{X}_f} = A \underbrace{\begin{bmatrix} x_0 & x_1 & \cdots x_{T-1} \end{bmatrix}}_{\mathbf{X}_p}. \tag{3-3}$$

[1] A *Dirac delta function* $\delta(k) = 1$ at $k = 0$ and $\delta(k) = 0 \ \forall \, k > 0$.

Equation (3-3) can be solved for A by

$$\bar{A} = \mathbf{X}_f \, \mathbf{X}_p^T \, (\mathbf{X}_p \, \mathbf{X}_p^T)^{-1}. \tag{3-4}$$

For sufficiently large T, the estimate $\bar{A} = A$ and hence the adversary can easily infer all the sensitive information about the model resulting in a privacy breach.

3-1-2 Existing Work and Our Contributions

There has been a large body of work done in the statistics and database literature on disclosure limitation and privacy-preserving publication of data [69, 70]. The recently proposed formulation of privacy by Dwork [71] called Differential Privacy (DP) has been adopted as a standard definition of privacy in many applications offering quantitative privacy guarantees. Originally, differential privacy was proposed for a static system as a measure of maximizing the accuracy of queries from statistical databases while minimizing the probability of identifying the individuals. In recent years, differential privacy has gained a significant amount of attention in the context of dynamical systems and control where researchers have used it for a diverse set of objectives such as control [72, 73], consensus [74–76], and optimization [77–84].

Although differential privacy has made its way to systems and control, very little work has been done in utilizing differential privacy for protecting the system model. To the best of our knowledge, only [64, 85, 86] discuss the problem of model-preservation in the DP framework. In [64], differential privacy was explored for designing output noises for preserving the model. The authors in [85] present several perturbation techniques to release a model describing the dynamics of a large group of users responding to a common single input signal and producing a single output signal. However, their approach assumes a *trusted* data aggregator where participants provide their confidential scalar model parameters to, and the aggregator then uses these parameters to publish a transfer function of the SISO system describing the relationship between common input and the aggregate output. In [86] differential privacy was used to protect the consensus network topology from an eavesdropper who may have an unauthorized access to the central estimator. They present a mechanism where each agent in the network adds DP-induced noise to its output and transmits it to the central estimator to estimate the topology matrix and its eigenvalues. However, the authors in [86] do not define any utility function and characterize the privacy-utility trade-offs. These gaps in the present state-of-the-art along with a strong fundamental need to protect the model motivates us to explore this problem in-depth using differential privacy.

The major contributions of this chapter are as follows:

- We propose the first differential privacy mechanism to study the problem of protecting the model privacy i.e., the system matrix A while releasing the state sequences for data aggregation *without* assuming an intermediary trusted aggregator. We also derive an analytical expression to estimate the minimum noise level required to guarantee differential privacy.
- Furthermore, we define a utility function in our problem setup and characterize the resulting privacy-utility trade-off using numerical illustrations. We also analyze the effect of the DP mechanism w.r.t the various privacy design parameters.

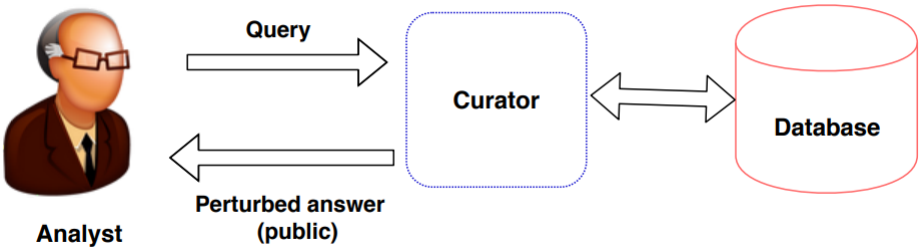


Figure 3-2: Original framework of differential privacy proposed by Dwork.

3-2 Technical Preliminaries

The basic idea behind differential privacy is to perturb the exact result before releasing it to the public. In the original DP framework proposed by Dwork [70] shown in Figure 3-2, there exists a curator who takes in the query of the data analyst and calculates the exact answer to the query by accessing the database. The curator then perturbs the result so that the output distribution over answers does not vary much if any given individual changes its data (or even participates or not) in a database. The amount of perturbation affects both the usefulness of the result and privacy. The more perturbation used, the less useful the result and a higher privacy is retained. Following are the some the definitions that are required to understand the mathematical underpinnings of differential privacy.

Definition 4. *(Data Base) [3]. A database D is a storage space containing structured set of data and user information that needs to be protected. Each element in a database corresponds to information from an individual user. The universe of all possible databases of interest is denoted by $\mathbb{D}$.*

Definition 5. *(Query) [3]. The quantity to be released to the public that we would like to compute from a database is modeled by $q(D)$ for some mapping q called query that acts on D.*

Definition 6. *(Adjacent Databases) [3]. Two databases $D = \{d_i\}$ and $D' = \{d'_i\}$ for $i = 1, 2, ...n$ are said to be adjacent if there exists $i \in \{1, ...n\}$ such that $d_j = d'_j$ for all $j \neq i$.*

Differential privacy is able to guarantee that the result of computation on a database does not change much when any single user in the database changes his/her information. As directly making $q(D)$ available to the public may cause users in the database to lose their privacy, a mechanism $\mathcal{M}$ is developed that approximates q. In the DP setting, all mechanisms under consideration are *randomized* and $\text{range}(M) = \text{range}(q)$. Below we present the definition of the differential privacy.

Definition 7. *(ϵ- Differential Privacy) [70]. Given $\epsilon \geq 0$, a mechanism $\mathcal{M}$ preserves ϵ-differential privacy if for all $R \subseteq \text{range}(\mathcal{M})$ and all adjacent databases D and D' in $\mathbb{D}$, it holds that:*

$$\Pr(\mathcal{M}(D) \in R) \leq e^{\epsilon} \Pr(\mathcal{M}(D') \in R). \tag{3-5}$$

The above definition implies that for a given database, the output of a differentially private mechanism obeys a certain probability distribution and acts on a database to ensure that two

adjacent databases are nearly indistinguishable (in a probabilistic sense) from just looking at the output of the mechanism. The probability measure in (3-5) is taken from the probability space for defining the randomized mechanism $\mathcal{M}$.

3-3 Problem Setup

3-3-1 Notation

$\|.\|_p$ represents the $p-$norm of a vector or the induced $p-$norm of a matrix with $p \in [1, \infty)$. If $x(k)$ represents the state vector at time instance k, then the state sequence up to time T is denoted by $\mathbf{X}[0:T]$. $Lap(0,b)^n$ denotes n-dimensional Laplace distribution with $i.i.d.$ components, each with a probability density function $p(x) = \frac{1}{2b}e^{-\frac{x}{b}}$.

3-3-2 Differential Privacy for System Model Identification

Differential privacy is a method of introducing randomness or noise into a particular system such that the adversary cannot uniquely identify the data to be protected while at the same time computing the query from the data with a considerable amount of utility [70]. In this case, the data to be protected is the model or the system matrix A while the query Q_k is

$$Q_k = x_k. \tag{3-6}$$

The noise is calibrated according to the sensitivity of the query and added to the state vectors x_k as given by (3-7). These perturbed samples will be transmitted to an external entity (potential adversary) who wants to compute the aggregate of all the state vectors up to time T.

$$\mathcal{M}_k : \qquad \tilde{x}_k = x_k + \eta_k, \tag{3-7}$$

To design the noise $\eta_k \in \mathbb{R}^n$ in $\mathcal{M}_k$ given by (3-7), we first present the following definitions:

Definition 8. *(β Adjacency) : Two state matrices A and A' are β adjacent (denoted by Adj^β) if for some $\beta \geq 0$,*

$$\mathrm{Adj}^\beta \stackrel{\mathrm{def}}{=} \left\|A - A'\right\|_2 \leq \beta \quad \forall \beta \geq 0. \tag{3-8}$$

Remark 1. *Adjacency in differential privacy captures the quantity to be hidden. Contrary to the popular definition of adjacency used in DP for static and dynamic systems [68, 85, 87], where adjacency is defined w.r.t. changes in only component i while keeping the other components $j \neq i$ unchanged, our definition allows changes that can possibly affect and change various components of the state matrix A.*

We borrowed this definition of adjacency from [86], where the authors used this adjacency relationship to protect the privacy of topology in consensus networks.

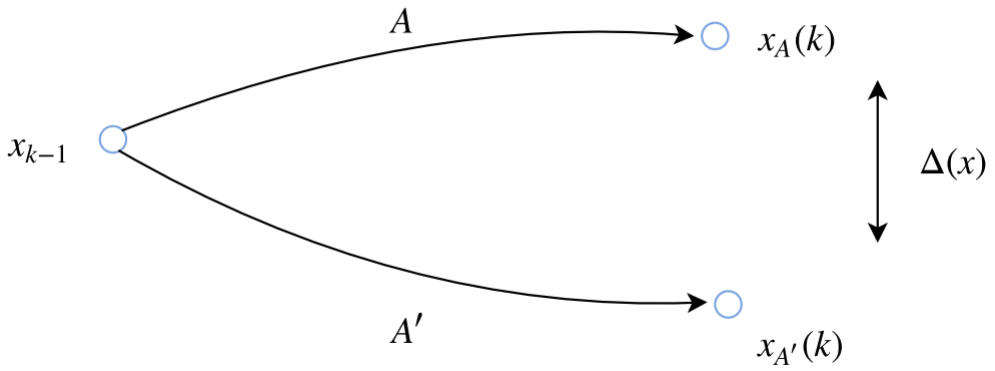


Figure 3-3: Illustration of the adjacency relationship and sensitivity.

Definition 9. *(Sensitivity) : Sensitivity $\Delta(x)$ represents the maximum possible difference between the two state vectors generated by any two β-adjacent state matrices starting from the same initial condition x_{k-1}. In this chapter, we define sensitivity in terms of the L_1 norm*

$$\begin{aligned} \Delta(x) &= \|x_A(k) - x_{A'}(k)\|_1, \\ &= \|Ax_{k-1} - A'x_{k-1}\|_1. \end{aligned} \tag{3-9}$$

Figure 3-3 illustrates the adjacency relationship and sensitivity.

Definition 10. *(Finite Time ϵ Differential Privacy): Given $\epsilon \geq 0$, the mechanism $\mathcal{M}$ given in* (3-10)

$$\mathcal{M}: \qquad \tilde{\boldsymbol{X}}[0:T] = \boldsymbol{X}[0:T] + \eta[0:T], \tag{3-10}$$

preserves ϵ-differential privacy up to time T if for any two β-adjacent state matrices A and A', and for any $R \subseteq \text{range}\,(\mathcal{M})$ the following relationship is satisfied:

$$\Pr\left[\tilde{\boldsymbol{X}}_A[0:T] \in R\right] \;\leq\; e^{\epsilon}\;\Pr\left[\tilde{\boldsymbol{X}}_{A'}[0:T] \in R\right]. \tag{3-11}$$

Definition 10 says that if the state matrix changes from A to A' that is β-adjacent, then the corresponding state trajectory statistics change atmost with a factor of e^{ϵ}, where ϵ quantifies the privacy loss.

Definition 11. *(Utility) : Utility $\mathcal{U}$ is defined as*

$$\mathcal{U} \;= 1 - \frac{\|X_{avg} - \tilde{X}_{avg}\|_1}{\|X_{avg}\|_1}, \tag{3-12}$$

where $X_{avg} = \frac{1}{T}\sum_{k=0}^{T} x_k$ and $\tilde{X}_{avg} = \frac{1}{T}\sum_{k=0}^{T} \tilde{x}_k$.

3-3-3 Adversarial Estimate

Upon receiving the perturbed state samples $\tilde{x}_k$, the adversary will be able to obtain an estimate denoted by $\hat{A}$ as follows:

$$\hat{A} \;= \underset{\tilde{A} \in R^{n \times n}}{\arg\min}\; \|\tilde{\mathbf{X}}_f - \tilde{A}\tilde{\mathbf{X}}_p\|, \tag{3-13}$$

where $\tilde{\mathbf{X}}_f = \tilde{\mathbf{X}}[1:T]$ and $\tilde{\mathbf{X}}_p = \tilde{\mathbf{X}}[0:T-1]$. The DP-mechanism in (3-7) will ensure that the state samples generated by A and $A' \in Adj^{\beta}$ are almost equally likely i.e., the state samples generated by Adj^{β} state matrices are *statistically not very different* [86]. Hence by observing these state samples, the adversary will not be able to distinguish between A and A' with high confidence level while retaining the necessary information to compute the query with reasonable utility. Thus, the privacy of the state matrix A is preserved which results in the estimation error

$$\mathbf{E} = \mathbb{E}\left[\left\|A - \hat{A}\right\|_2\right]. \tag{3-14}$$

The term $\left\|A - \hat{A}\right\|_2$ is defined as the perturbation norm.

3-3-4 Problem Statement

Given the above problem setup: design the noise $\eta(k)$ in (3-7) satisfying the DP-definition in (3-11) for a given ϵ, β, and characterize the resulting trade-off between the level of privacy and the utility.

3-4 Proposed Solution

In this section, we present a noise adding DP-mechanism to protect the privacy of the system matrix A. The most common way to implement the DP mechanism is to add noise generated according to the Laplacian distribution based on the sensitivity Δ of the system [70]. To design $\eta(k)$, we first need to estimate the sensitivity up to time T given as:

$$\Delta(T) \;=\; \max_{A, A' : Adj^{\beta}} \left\|\mathbf{X}_A[0:T] - \mathbf{X}_{A'}[0:T]\right\|_1. \tag{3-15}$$

Once $\Delta(T)$ is obtained, the following theorem provides a sufficient condition for the noise design.

Theorem 1. *[87]: The mechanism $\mathcal{M}$ in (3-10) is ϵ- differentially private up to time T if $\eta(k)$ is white Laplacian noise with distribution $\eta(k) \sim Lap(0,b)^n$ and $b \geq \frac{\Delta(T)}{\epsilon}$.*

Proof. The proof follows from Theorem 2 in [87]. Refer Appendix 4-2 for a detailed proof. □

Remark 2. *Notice that the noise design parameter b is inversely proportional to ϵ. Thus as ϵ decreases, the noise parameter b increases resulting in a flat tail Laplacian distribution curve i.e., the probability of picking a random number close to zero is very low and hence a higher noise level is generated. As a result, when ϵ decreases, the privacy level increases and vice-versa. Also, notice the noise parameter b is directly proportional to the sensitivity $\Delta(T)$. Thus the lower the sensitivity, the lower the noise that needs to be added and vice-versa.*

Intuitively, if the sensitivity is low, then for two different $\beta-$adjacent state matrices, the change in the corresponding state trajectories will not be large, and hence the level of noise required to make the two-state trajectories *statistically not very different* will also be small

[86]. Since sensitivity is crucial in designing a DP-induced noise, we may try to calculate $\Delta(T)$. However, it is difficult to obtain an analytical expression for $\Delta(T)$. Hence, we propose the following theorem instead that characterizes the upper bound on the sensitivity. Through this upper bound, we obtain the minimum noise level required to ensure DP.

Theorem 2. *The sensitivity $\Delta(T)$ is upper bounded by*

$$\Delta(T) \;\leq\; \sqrt{n}\,\beta\,\|x(0)\|_1 \sum_{k=0}^{T} \|A^k\|_1 \tag{3-16}$$

Proof. To obtain the upper bound for the sensitivity function $\Delta(T)$ given by (3-15), let us consider two measurements x_A and x'_A produced by β adjacent state matrices A and A'

$$\begin{aligned}\Big\|x_A(k+1) - x_{A'}(k+1)\Big\|_1 &= \|Ax(k) - A'x(k)\|_1 \\ &\leq \|A - A'\|_1 \,\|x(k)\|_1 \\ &\leq \sqrt{n}\|A - A'\|_2 \,\|x(k)\|_1 \\ &\leq \sqrt{n}\,\beta\,\|x(0)\|_1 \|A^k\|_1\end{aligned}$$

$$\begin{aligned}\Big\|\mathbf{X}_A[0:T] - \mathbf{X}_{A'}[0:T]\Big\|_1 &= \sum_{k=0}^{T} \Big\|x_A(k+1) - x_{A'}(k+1)\Big\|_1 \\ &\leq \sqrt{n}\,\beta\,\|x(0)\|_1 \sum_{k=0}^{T} \|A^k\|_1\end{aligned} \tag{3-17}$$

□

From (3-17), we can see that the sensitivity is bounded for a finite value of T. Also, when A is a Schur matrix (eigenvalues inside the unit disc), the sensitivity is bounded for all values of T. The noise $\eta(k)$ can be generated by setting the noise design parameter b as follows

$$\begin{aligned}b &= \frac{\Delta(T)}{\epsilon} \\ &= \sqrt{n}\;\frac{\beta}{\epsilon}\,\|x(0)\,\|_1 \sum_{k=0}^{T} \|A^k\|\end{aligned} \tag{3-18}$$

Note that in (3-18), β and ϵ are the privacy design parameters and the ratio

$$\lambda \stackrel{\text{def}}{=} \frac{\beta}{\epsilon} \tag{3-19}$$

represents the *privacy level* [86] whereas β and ϵ are known as the privacy design parameters. If the adjacency parameter β increases for a fixed ϵ, it indicates that DP is ensured for a larger set of state matrices. Similarly if ϵ decreases, it denotes the increase in privacy level.

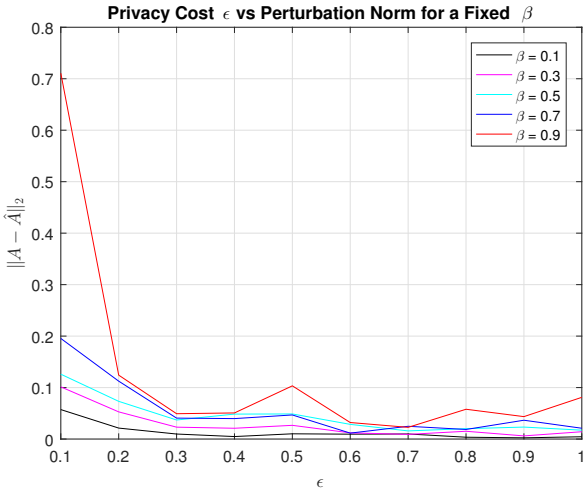


Figure 3-4: Illustration of the DP Mechanism simulated for various values of β and ϵ

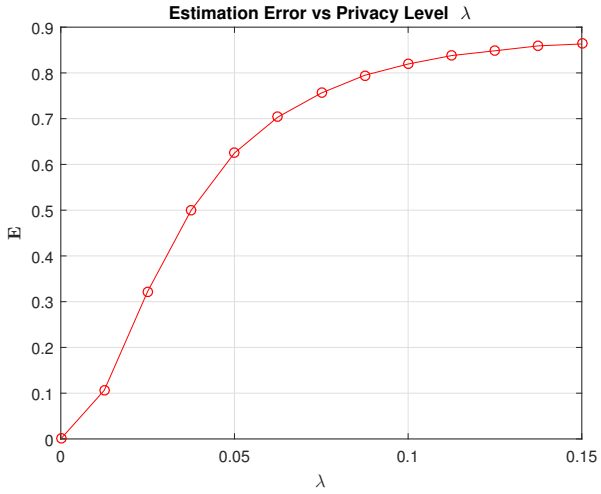


Figure 3-5: State matrix estimation error.

3-5 Simulation Results

We consider the same supply chain example explained in Section 3-1-1. Let the system matrix A be

$$A = \begin{bmatrix} 0.16 & 0 & 0 \\ 0.8 & 0.25 & 0.01 \\ 0 & 0.7 & 0.19 \end{bmatrix}.$$

We set $T = 15$ days and $x_0 = \begin{bmatrix} 1000 & 0 & 0 \end{bmatrix}^T$.

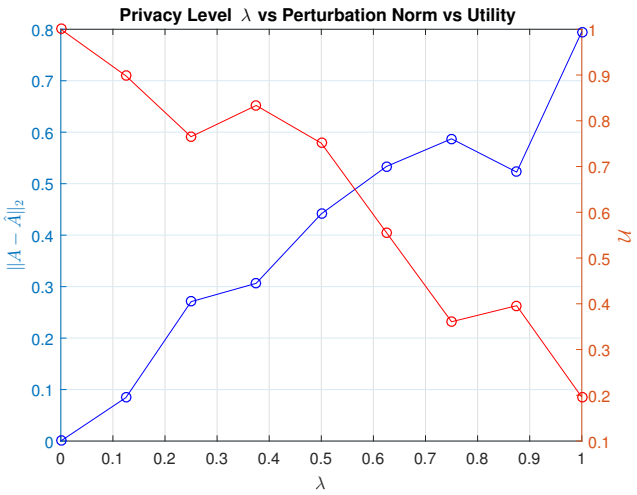


Figure 3-6: Privacy-Utility trade-off characterization

3-5-1 System Matrix Estimation

The adversary obtains an estimate of the system matrix $\hat{A}$ given by (3-13). Figure 3-4 shows the perturbation norm $\|A - \hat{A}\|_2$ simulated for various values of β and ϵ. It is clear that as ϵ increases the perturbation norm tends to zero i.e., the adversarial estimate approaches the true value, and hence there is no privacy. It is also evident that as β increases for a fixed ϵ, DP is ensured for a larger set of state matrices. It is seen that when β is increased, the noise level added is also increased because of the increase in sensitivity level. Hence a larger difference in the perturbation norm is observed for higher β's.

Next, we simulated for multiple noise realizations and calculated the sample mean to approximate the expected error $\mathbf{E}$ given in (3-14). Figure 3-5 shows the variation of the expected error w.r.t.the privacy level λ. Naturally, when $\lambda = 0$ i.e., no privacy, the expected error $\mathbf{E} = 0$, and as expected, the estimation accuracy degrades with an increase in the privacy level.

3-5-2 Privacy vs Utility

We now characterize the privacy-utility trade-off by calculating the perturbation norm and utility $\mathcal{U}$ for various values of the privacy level λ. Figure 3-6 depicts the plot between the perturbation norm and utility for various values of the privacy level λ. This plot aids in choosing the privacy and utility level for this system. For example, when $\lambda = 0.5$ we have,

$$X_{avg} = \begin{bmatrix} 73.4361 \\ 127.2418 \\ 62.0275 \end{bmatrix}, \quad \tilde{X}_{avg} = \begin{bmatrix} 89.9353 \\ 119.3256 \\ 85.5776 \end{bmatrix}, \quad \mathcal{U} = 0.76.$$

$$\hat{A} = \begin{bmatrix} 0.0758 & 0.0922 & 0.1149 \\ 0.6571 & 0.3963 & 0.0599 \\ -0.0650 & 0.3919 & -0.0501 \end{bmatrix}, \quad \|A - \hat{A}\|_2 = 0.45.$$

Notice the difference between the actual A matrix and the matrix $\hat{A}$ obtained by the adversary from the perturbed state trajectories. As emphasized in Remark 1, our adjacency definition allowed changes in multiple components of the A matrix. Let us now compare both the matrices element-wise. From A we see that $a_{12} = 0$, which means in the actual supply chain model, the producer P does not return any percentage of raw materials back to the supplier S, whereas $\hat{a}_{12} \sim 0.1$ which misleads the adversary to think that about 10% of raw materials received from the supplier is returned back to the supplier possibly to defects. The adversary certainly cannot deduce with high confidence level if the producer actually returned some fraction of raw materials to the supplier, thereby, preserving the privacy of the model. Similarly, in the actual model $a_{13} = 0$ and $a_{31} = 0$ i.e., there is no communication between supplier and retailer, whereas the components $\hat{a}_{13}$ and $\hat{a}_{31} \neq 0$.

Analyzing the eigenvalues of A, we have:

$$\gamma = \begin{bmatrix} 0.1311 & 0.3089 & 0.1600 \end{bmatrix}^T,$$

whereas the adversarial estimated eigenvalues are:

$$\hat{\gamma} = \begin{bmatrix} 0.6188 & -0.0984 + 0.2047i & -0.0984 - 0.2047i \end{bmatrix}^T.$$

The adversary here is again misled to think that the model exhibits oscillatory behavior due to complex poles, whereas, no such oscillatory behavior is present in the original system. Thus, through differential privacy, we are able to mislead the adversary in several directions while still managing to retain a high level of utility ($\mathcal{U} \approx 0.8$).

3-6 Conclusions

In this chapter, we proposed a differential privacy mechanism to protect the system matrix. The proposed mechanism adds synthetic noise generated according to the Laplacian probability distribution and prevents an external adversary from uniquely identifying the system matrix when accessing the state samples to compute the aggregate (average) information. We derived an analytical bound on the sensitivity function and calculated the sufficient noise level required to ensure DP. Simulation results validate the DP theory and show that the expected estimation error increases with an increase in the privacy level. Furthermore, we characterized the resulting trade-off between the privacy level and utility through empirical evidence. Using this characterization, we inferred how differential privacy aids in the process of introducing uncertainty and ambiguity in the adversarial mind while still retaining higher levels of utility. Deploying differential privacy in practice requires no major computational overhead since the DP-noise generation process is equivalent to generating random numbers according to a given probability distribution and sensitivity. Hence, a DP mechanism can be implemented easily provided it is feasible to obtain an analytical expression for the upper bound of the sensitivity function of a given query.

Chapter 4

Future Work and Overall Conclusions

In this chapter, we state the future research directions for both Chapter 2 and 3 and present the overall conclusions of this thesis.

4-1 Future Work

4-1-1 Chapter 2

In Chapter 2, we proposed $Obfuscate(.)$ - an obfuscation scheme to preserve the privacy of user behavioral patterns and the power network topology from the malicious cloud. Although the behavioral pattern and the power dynamics of the other smart meters in every locality are hidden from the malicious cloud, the respective lead meter can still infer this information. This is because the lead meter has access to the scaled measurements $z' = a_{ij}.z$ (Pearson coefficient = 1) whose dynamics are exactly the same as z. Hence, it was essential in our problem to consider a single non-collusive trusted node in every local network termed as the lead meter to initiate the obfuscation of power measurement dynamics. Future work may involve developing security protocols without assuming even a single trusted node in the network.

Another drawback of $Obfuscate(.)$ is that it does not take into account the grid configuration matrix $\mathbf{H}$ although time-invariant during state estimation may still be susceptible to changes all the time. For example, consider a person living in a particular locality is now motivated to install a smart meter at his home due to good security reasons or a person living in one locality is now moving to another locality. Such situations clearly result in the change in the network topology (an extra row addition or row deletion of the existing $\mathbf{H}$ matrix), and assuming a pre-computation of $\mathbf{H}^+$ at every stage is not reasonable. Hence, to deal with such instances, we require a protocol computing the matrix $A = (\mathbf{H}^T\mathbf{H})^{-1}$ for secure outsourcing of large matrix inversion to the cloud. During the course of this thesis, we designed another obfuscation-based protocol $MatrixInvert(.)$ given in Algorithm 9 that could possibly carry out a large matrix inversion in the cloud with $O(n^2)$ computational overhead to the client

Algorithm 9 $MatrixInvert(A)$

Input $A \in \mathbb{R}^{n\times n}$
Generate a uniform diagonal random matrix $R \in \mathbb{R}^{n\times n}$
Client ($\mathcal{U}$) computes the matrix $M = AR$ ▷ $O(n^2)$ as R is diagonal.
Send M to the cloud for matrix inversion.
Cloud computes $M^{-1} = (AR)^{-1} = R^{-1}A^{-1}$ ▷ $O(n^3)$ for matrix inversion.
Cloud returns $M^{-1} = R^{-1}A^{-1}$ to the client for verification.
Client generates a random vector $v \in \mathbb{R}^n$ and checks
if $(M^{-1}.(AR).v) == R^{-1}A^{-1}((AR).v) == v$ **then** ▷ positive verification
    Client computes the matrix $A' = R(R^{-1}A^{-1}) = A^{-1}$ ▷ $O(n)$ as R is diagonal
else ▷ negative verification.
    abort.
end if
Output: A^{-1}

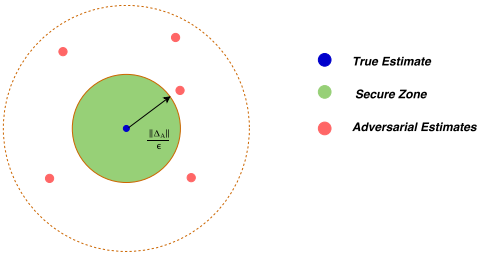
($\mathcal{U}$) while the cloud performs a computation of $O(n^3)$. The proof of verification, security, and correctness of this protocol follows from a similar approach to the one employed for $Obfuscate(.)$. However, the problem with this $MatrixInvert(.)$ is that it does not hide the sparsity pattern of the grid configuration when taking inverses i.e, it does not preserve the privacy of zero elements in the **H** matrix, thereby, making the power network vulnerable to stealth false data injection attacks. This raises a fundamental question in cryptography of how to protect the privacy of the zero elements of the matrix while outsourcing a large matrix inversion problem to a malicious cloud server? To the best of our knowledge, the existing literature does not have a solution to this problem.

Another work may involve developing a statistical measure to quantify the degree of obscurity introduced by these obfuscation schemes to understand how indistinguishable the obfuscated datasets are compared to the original measurements.

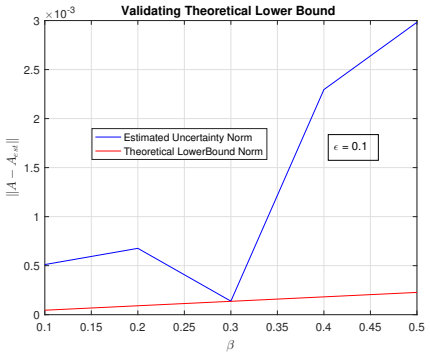
4-1-2 Chapter 3

In Chapter 3, we proposed a DP framework to protect the system matrix of an autonomous system model. Future work may involve extending this framework to a more generic non-autonomous systems with the privacy goal of protecting both the system and input matrices. Another research direction may involve designing an asymptotically decaying noise to retain higher levels of utility and providing mathematical proofs that the resulting mechanism preserves differential privacy.

It is also important to note that, we tracked and modeled only the movement of goods without considering the *pricing effect* in the supply chain model. An interesting project could be to track down and model the movement of prices at each stage of the supply chain process in the control theory framework and investigate the privacy concerns arising in such models. Modeling economic systems as complex dynamic systems in the control theory framework has now received a significant attention to a better understanding of economic phenomena. The amount of information economies create has grown exponentially and this line of research is now popularly known as complexity economics [8].



(a) Visual depiction of the Lemma



(b) Numerical illustration of the Lemma

Figure 4-1: Minimum model error achieved through DP as per the derived Lemma

Another interesting line of research and an open problem as pointed out in [85] could focus on analytically quantifying the approximation error and exploring the lower bounds on the model error achievable by DP mechanisms. During this project, we attempted to analytically quantify the lower bounds on the model error by considering adjacency relationship with respect to the state vectors rather than the system model (see. 3-8) as in Chapter 3. We arrived at a lower bound for the minimum amount of uncertainty introduced in the model when perturbing the state trajectories using differential privacy. The reader is referred to Appendix -3 for more details on this problem. Figure4-1(a) shows the visual representation of the problem, we are trying to solve and Figure 4-1(b) shows a tight lower bound obtained from the derived Lemma for a simple second order system (see. Appendix -3). However, when simulated for systems of higher order, we noticed that the expressions derived for lower bound were rather conservative and at times, our simulations even violated the minimum theoretical bound derived. Thus, this requires further investigation to develop a tighter lower bound for quantifying the minimum model error achievable through DP.

4-2 Overall Conclusions

In this thesis, we investigated the privacy concerns arising in dynamical systems in the context of state estimation - one of the most widely studied concepts in systems and control and relied on cryptography to address these privacy challenges. We studied two different problems with applications to smart grids and supply chain economics. To address the privacy challenges in smart grids, we deployed $Obfuscate(.)$ - a randomization scheme based on the

obfuscated transformation that is easier and efficient to implement in a smart meter platform. $Obfuscate(.)$ was able to guarantee the accurate estimate of the state variables with proper security, correctness, verification, and efficiency. In the case of supply chain economics, where protecting the privacy of a system model becomes crucial for a successful business, we deployed differential privacy - a noise addition technique. The resulting mechanism was able to achieve higher levels of utility while ensuring acceptable privacy level to mislead the adversary in several directions. Furthermore, we pointed out that the development of complexity economics to study economic phenomena in the control theory framework can also increase the privacy challenges of economics systems since *system identification* techniques can be used to reverse engineer the sensitive information of the model.

As the digital transformation is accelerating with many recent technological advancements such as blockchain, machine learning, etc., there is a growing importance of multi-disciplinary research both in academia and industry. The main goal of this project was to take a step towards a joint project proposal to bridge the gap between the systems and control and cyber security community. We hope that our work will serve as a motivation for future research collaborations between the two communities.

Bibliography

[1] R. Deng, "Why we need to improve cloud computing's security?." https://phys.org/news/2017-10-cloud.html, October 2017.

[2] R. Deng, G. Xiao, and R. Lu, "Defending against false data injection attacks on power system state estimation," *IEEE Transactions on Industrial Informatics*, vol. 13, pp. 198–207, Feb 2017.

[3] J. Cortes, G. E. Dullerud, S. Han, J. L. Ny, S. Mitra, and G. J. Pappas, "Differential privacy in control and network systems," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 4252–4272, Dec 2016.

[4] U.S District Court, District of Hawaii, Honolulu, HI,USA, *Adam Asquith vs Kauai Island Utility Co-operative*, June 2012.

[5] The Irish Times, "Facebook to contact 87 million users affected by data breach." https://www.irishtimes.com/business/technology/facebook-data-breach-affected-up-to-87-million-users-1.3450735, April 2018.

[6] P. Venkitasubramaniam, J. Yao, and P. Pradhan, "Information-theoretic security in stochastic control systems," *Proceedings of the IEEE*, vol. 103, pp. 1914–1931, Oct 2015.

[7] G. T. Duncan and D. Lambert, "Disclosure-limited data dissemination," *Journal of the American Statistical Association*, vol. 81, no. 393, pp. 10–18, 1986.

[8] O. Criner, "Control systems identification in finance and economics," *WIT Transactions on Information and Communication Technologies*, vol. 41, pp. 3–12, 2008.

[9] European Commission, "Energy. Smart grids and meters." https://ec.europa.eu/energy/en/topics/markets-and-consumers/smart-grids-and-meters, 2017.

[10] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin, "Private memoirs of a smart meter," in *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems*

for Energy-Efficiency in Building, BuildSys '10, (New York, USA), pp. 61–66, ACM, 2010.

[11] M. A. Lisovich, D. K. Mulligan, and S. B. Wicker, "Inferring personal information from demand-response systems," *IEEE Security Privacy*, vol. 8, pp. 11–20, Jan 2010.

[12] K. Kursawe, G. Danezis, and M. Kohlweiss, "Privacy-friendly aggregation for the smart-grid," in *Privacy Enhancing Technologies* (S. Fischer-Hübner and N. Hopper, eds.), pp. 175–191, Springer Berlin Heidelberg, 2011.

[13] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," in *2011 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 239–240, Jan 2011.

[14] Y. Liu, M. K. Reiter, and P. Ning, "False data injection attacks against state estimation in electric power grids," *ACM Trans. Inf. Syst. Secur.*, vol. 14, pp. 13:1–13:33, 2009.

[15] Y. F. Huang, S. Werner, J. Huang, N. Kashyap, and V. Gupta, "State estimation in electric power grids: Meeting new challenges presented by the requirements of the future grid," *IEEE Signal Processing Magazine*, vol. 29, pp. 33–43, Sept 2012.

[16] M. A. Rahman and G. K. Venayagamoorthy, "Distributed dynamic state estimation for smart grid transmission system," *IFAC-PapersOnLine*, vol. 50, no. 2, pp. 98 – 103, 2017. Control Conference Africa CCA 2017.

[17] Y. Kim, E. C. H. Ngai, and M. B. Srivastava, "Cooperative state estimation for preserving privacy of user behaviors in smart grid," in *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 178–183, Oct 2011.

[18] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," *IEEE Internet Computing*, vol. 16, pp. 69–73, Jan. 2012.

[19] C. Wang, K. Ren, and J. Wang, "Secure and practical outsourcing of linear programming in cloud computing," in *2011 Proceedings IEEE INFOCOM*, pp. 820–828, April 2011.

[20] C. Gentry, *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009.

[21] Z. Erkin, "Private data aggregation with groups for smart grids in a dynamic setting using CRT," in *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, Nov 2015.

[22] S. Ge, P. Zeng, R. Lu, and K.-K. R. Choo, "FGDA: Fine-grained data analysis in privacy-preserving smart grid communications," *Peer-to-Peer Networking and Applications*, Nov 2017.

[23] F. Knirsch, D. Engel, and Z. Erkin, "A fault-tolerant and efficient scheme for data aggregation over groups in the smart grid," in *IEEE Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, Dec 2017.

[24] K. Emura, "Privacy-preserving aggregation of time-series data with public verifiability from simple assumptions," in *Information Security and Privacy* (J. Pieprzyk and S. Suriadi, eds.), pp. 193–213, Springer International Publishing, 2017.

[25] G. Danezis, C. Fournet, M. Kohlweiss, and S. Zanella-Béguelin, "Smart meter aggregation via secret-sharing," in *Proceedings of the First ACM Workshop on Smart Energy Grid Security*, SEGS '13, (New York, NY, USA), pp. 75–80, ACM, 2013.

[26] M. A. Rahman and H. Mohsenian-Rad, "False data injection attacks against nonlinear state estimation in smart power grids," in *2013 IEEE Power Energy Society General Meeting*, pp. 1–5, July 2013.

[27] A. Beussink, K. Akkaya, I. F. Senturk, and M. M. E. A. Mahmoud, "Preserving consumer privacy on IEEE 802.11s-based smart grid ami networks using data obfuscation," in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 658–663, April 2014.

[28] S. Tonyali, O. Cakmak, K. Akkaya, M. M. E. A. Mahmoud, and I. Guvenc, "Secure data obfuscation scheme to enable privacy-preserving state estimation in smart grid ami networks," *IEEE Internet of Things Journal*, vol. 3, pp. 709–719, Oct 2016.

[29] C. Efthymiou and G. Kalogridis, "Smart grid privacy via anonymization of smart metering data," in *2010 First IEEE International Conference on Smart Grid Communications*, pp. 238–243, Oct 2010.

[30] F. Li, B. Luo, and P. Liu, "Secure information aggregation for smart grids using homomorphic encryption," in *2010 First IEEE International Conference on Smart Grid Communications*, pp. 327–332, Oct 2010.

[31] F. C. Schweppe and J. Wildes, "Power system static-state estimation, part I: Exact model," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-89, pp. 120–125, Jan 1970.

[32] F. C. Schweppe and D. B. Rom, "Power system static-state estimation, part II: Approximate model," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-89, pp. 125–130, Jan 1970.

[33] F. C. Schweppe, "Power system static-state estimation, part iii: Implementation," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-89, pp. 130–135, Jan 1970.

[34] I. Gera, Y. Yakoby, and T. Routtenberg, "Blind estimation of states and topology (BEST) in power systems," in *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 1080–1084, Nov 2017.

[35] G. Liang, J. Zhao, F. Luo, S. R. Weller, and Z. Y. Dong, "A review of false data injection attacks against modern power systems," *IEEE Transactions on Smart Grid*, vol. 8, pp. 1630–1638, July 2017.

[36] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "Matpower's extensible optimal power flow architecture," in *2009 IEEE Power Energy Society General Meeting*, pp. 1–7, July 2009.

[37] A. Wood and B. Wollenberg, *Power Generation, Operation, and Control*. A Wiley-Interscience publication, Wiley, 1996.

[38] Y. F. Huang, S. Werner, J. Huang, N. Kashyap, and V. Gupta, "State estimation in electric power grids: Meeting new challenges presented by the requirements of the future grid," *IEEE Signal Processing Magazine*, vol. 29, pp. 33–43, Sept 2012.

[39] O. Krause and S. Lehnhoff, "Generalized static-state estimation," in *2012 22nd Australasian Universities Power Engineering Conference (AUPEC)*, pp. 1–6, Sept 2012.

[40] U. S. Department of Energy, "Factors affecting PMU installation costs." https://www.smartgrid.gov/files/PMU-cost-study-final-10162014_1.pdf, October 2014.

[41] M. Cosovic and D. Vukobratovic, "Fast real-time DC state estimation in electric power systems using belief propagation," *Computing Research Repository, CoRR*, vol. abs/1705.01376, 2017.

[42] A. Monticelli, "Electric power system state estimation," *Proceedings of the IEEE*, vol. 88, pp. 262–282, Feb 2000.

[43] "U.S.-Canada power system outage task force.." https://digital.library.unt.edu/ark:/67531/metadc26005/, August 2003.

[44] S. A. Salinas and P. Li, "Privacy-preserving energy theft detection in microgrids: A state estimation approach," *IEEE Transactions on Power Systems*, vol. 31, pp. 883–894, March 2016.

[45] Y. Shoukry, K. Gatsis, A. Alanwar, G. J. Pappas, S. A. Seshia, M. Srivastava, and P. Tabuada, "Privacy-aware quadratic optimization using partially homomorphic encryption," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 5053–5058, Dec 2016.

[46] Y. Lindell and B. Pinkas, "Secure Multi-party Computation for Privacy-Preserving Data Mining," *Journal of Privacy and Confidentiality*, vol. 1, no. 1, pp. 59–98, 2009-10.

[47] European Commission, "Benchmarking smart metering deployment in the EU-27 with a focus on electricity." https://ses.jrc.ec.europa.eu/publications/reports/benchmarking-smart-metering-deployment-eu-27-focus-electricity, March 2017.

[48] G. Hunt, "What does GDPR mean for your energy business?." https://www.siliconrepublic.com/enterprise/gdpr-energy-sector, March 2017.

[49] M. Tebaa and S. E. Hajji, "Secure cloud computing through homomorphic encryption," *Computing Research Repository - CoRR*, vol. abs/1409.0829, 2014.

[50] M. Simos, "Microsoft security intelligence report.." https://www.microsoft.com/en-us/security/Intelligence-report, March 2017.

[51] M. J. Atallah and K. B. Frikken, "Securely outsourcing linear algebra computations," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '10, (New York, NY, USA), pp. 48–59, ACM, 2010.

[52] M. J. Atallah, K. B. Frikken, and S. Wang, "Private outsourcing of matrix multiplication over closed semi-rings," in *SECRYPT*, 2012.

[53] D. Fiore and R. Gennaro, "Publicly verifiable delegation of large polynomials and matrix computations, with applications," in *Proceedings of the Conference on Computer and Communications Security*, CCS '12, (New York, NY, USA), pp. 501–512, ACM, 2012.

[54] Y. Zhang and M. Blanton, "Efficient secure and verifiable outsourcing of matrix multiplications," in *Information Security* (S. S. M. Chow, J. Camenisch, L. C. K. Hui, and S. M. Yiu, eds.), pp. 158–178, Springer International Publishing, 2014.

[55] M. Kumar, J. Meena, and M. Vardhan, "Privacy preserving, verifiable and efficient outsourcing algorithm for matrix multiplication to a malicious cloud server," *Cogent Engineering*, vol. 4, no. 1, 2017.

[56] J. Dreier and F. Kerschbaum, "Practical privacy-preserving multiparty linear programming based on problem transformation," in *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pp. 916–924, Oct 2011.

[57] A. López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," in *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, (New York, NY, USA), pp. 1219–1234, ACM, 2012.

[58] J. Saia and M. Zamani, "Recent results in scalable multi-party computation," in *SOFSEM 2015: Theory and Practice of Computer Science* (G. F. Italiano, T. Margaria-Steffen, J. Pokorný, J.-J. Quisquater, and R. Wattenhofer, eds.), (Berlin, Heidelberg), pp. 24–44, Springer Berlin Heidelberg, 2015.

[59] S. Goldwasser, Y. T. Kalai, and G. N. Rothblum, "Delegating computation: Interactive proofs for muggles," *J. ACM*, vol. 62, pp. 27:1–27:64, Sept. 2015.

[60] X. Lei, X. Liao, T. Huang, H. Li, and C. Hu, "Outsourcing large matrix inversion computation to a public cloud," *IEEE Transactions on Cloud Computing*, vol. 1, pp. 1–1, Jan 2013.

[61] F. Chen, J. Dai, B. Wang, S. Sahu, M. Naphade, and C.-T. Lu, "Activity analysis based on low sample rate smart meters," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, (New York, NY, USA), pp. 240–248, ACM, 2011.

[62] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, pp. 12–19, Feb 2011.

[63] H. Tohidi and M. M. Jabbari, "Innovation as a success key for organizations," *Procedia Technology*, vol. 1, pp. 560 – 564, 2012. First World Conference on Innovation and Computer Sciences (INSODE 2011).

[64] G. Bottegal, F. Farokhi, and I. Shames, "Preserving privacy of finite impulse response systems," *IEEE Control Systems Letters*, vol. 1, pp. 128–133, July 2017.

[65] X. He, F. Zhang, and N. Adam, "Towards ranking the importance of patents," in *2008 IEEE Symposium on Advanced Management of Information for Globalized Enterprises (AMIGE)*, pp. 1–5, Sept 2008.

[66] G. Pillonetto, "The interplay between system identification and machine learning," *Computing Research Repository CoRR*, vol. abs/1612.09158, 2016.

[67] S. Saitta, B. Raphael, and I. F. C. Smith, "Combining two data mining methods for system identification," in *Intelligent Computing in Engineering and Architecture* (I. F. C. Smith, ed.), (Berlin, Heidelberg), pp. 606–614, Springer Berlin Heidelberg, 2006.

[68] L. Fan and L. Xiong, "An adaptive approach to real-time aggregate monitoring with differential privacy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, pp. 2094–2106, Sept 2014.

[69] G. T. Duncan and D. Lambert, "Disclosure-limited data dissemination," *Journal of the American Statistical Association*, vol. 81, no. 393, pp. 10–18, 1986.

[70] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, pp. 211–407, Aug. 2014.

[71] C. Dwork, "Differential privacy," in *Automata, Languages and Programming* (M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, eds.), (Berlin, Heidelberg), pp. 1–12, Springer Berlin Heidelberg, 2006.

[72] Z. Huang, Y. Wang, S. Mitra, and G. E. Dullerud, "On the cost of differential privacy in distributed control systems," in *Proceedings of the 3rd International Conference on High Confidence Networked Systems*, HiCoNS '14, (New York, NY, USA), pp. 105–114, ACM, 2014.

[73] Y. Wang, Z. Huang, S. Mitra, and G. E. Dullerud, "Entropy-minimizing mechanism for differential privacy of discrete-time linear feedback systems," in *53rd IEEE Conference on Decision and Control*, pp. 2130–2135, Dec 2014.

[74] Z. Huang, S. Mitra, and G. Dullerud, "Differentially private iterative synchronous consensus," in *Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society*, WPES '12, (New York, NY, USA), pp. 81–90, ACM, 2012.

[75] E. Nozari, P. Tallapragada, and J. Cortes, "Differentially private average consensus: Obstructions, trade-offs, and optimal algorithm design," *Automatica*, vol. 81, pp. 221 – 231, 2017.

[76] Y. Mo and R. M. Murray, "Privacy preserving average consensus," *IEEE Transactions on Automatic Control*, vol. 62, pp. 753–765, Feb 2017.

[77] S. Han, U. Topcu, and G. J. Pappas, "Differentially private distributed constrained optimization," *IEEE Transactions on Automatic Control*, vol. 62, pp. 50–64, Jan 2017.

[78] Z. Huang, S. Mitra, and N. Vaidya, "Differentially private distributed optimization," in *Proceedings of the 2015 International Conference on Distributed Computing and Networking*, ICDCN '15, (New York, NY, USA), pp. 4:1–4:10, ACM, 2015.

[79] M. T. Hale and M. Egerstedty, "Differentially private cloud-based multi-agent optimization with constraints," in *2015 American Control Conference (ACC)*, pp. 1235–1240, July 2015.

[80] E. Nozari, P. Tallapragada, and J. CortÃŀs, "Differentially private distributed convex optimization via objective perturbation," in *2016 American Control Conference (ACC)*, pp. 2061–2066, July 2016.

[81] S. Han, U. Topcu, and G. J. Pappas, "Differentially private convex optimization with piecewise affine objectives," in *53rd IEEE Conference on Decision and Control*, pp. 2160–2166, Dec 2014.

[82] J. Hsu, A. Roth, T. Roughgarden, and J. Ullman, "Privately solving linear programs," *Computing Research Repository CoRR*, vol. abs/1402.3631, 2014.

[83] Q. Ling, W. Xu, M. Wang, and Y. Li, *Distributed Constrained Optimization Over Cloud-Based Multi-agent Networks*, pp. 91–102. Cham: Springer International Publishing, 2016.

[84] E. Nozari, P. Tallapragada, and J. Cortes, "Differentially private distributed convex optimization via functional perturbation," *IEEE Transactions on Control of Network Systems*, vol. PP, no. 99, pp. 1–1, 2016.

[85] J. Le Ny and G. J. Pappas, "Privacy-preserving release of aggregate dynamic models," in *Proceedings of the 2Nd ACM International Conference on High Confidence Networked Systems*, HiCoNS '13, (New York, NY, USA), pp. 49–56, ACM, 2013.

[86] V. Katewa, A. Chakrabortty, and V. Gupta, "Protecting privacy of topology in consensus networks," in *2015 American Control Conference (ACC)*, pp. 2476–2481, July 2015.

[87] J. L. Ny and G. J. Pappas, "Differentially private filtering," *IEEE Transactions on Automatic Control*, vol. 59, pp. 341–354, Feb 2014.

[88] J. D. McCalley, "The Power Flow Problem," tech. rep., Iowa State University, June 2018.

Appendix

-1 A fully measured 5-bus power system

A fully measured 5-bus power system is shown in Figure 2. In this case, the total number of meters m is 10 and the meter measurements are $z = [F_{12}, F_{23}, F_{24}, F_{35}, F_{45}, P_1, P_2, P_3, P_4, P_5]^T$ where F_{ij} represents the branch (i, j) active power flow and P_j represents bus j active power injection. The structure of the measurement matrix $\mathbf{H}$ is then given by [2]:

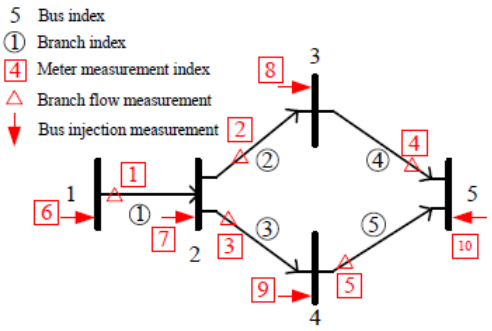


Figure 2: A fully measured 5-bus power system. Taken from [2]

$$H = \begin{pmatrix} b_{12} & 0 & 0 & 0 \\ -b_{23} & b_{23} & 0 & 0 \\ -b_{24} & 0 & b_{24} & 0 \\ 0 & -b_{35} & 0 & b_{35} \\ 0 & 0 & -b_{45} & b_{45} \\ b_{12} & 0 & 0 & 0 \\ -b_{12} - b_{23} - b_{24} & b_{23} & b_{24} & 0 \\ b_{23} & -b_{23} - b_{35} & 0 & b_{35} \\ b_{24} & 0 & -b_{24} - b_{45} & b_{45} \\ 0 & -b_{35} & -b_{45} & b_{35} + b_{45} \end{pmatrix}$$

where b_{ij} denotes the susceptance of the transmission line (i, j). The susceptance is the imaginary part of admittance and the admittance matrix is obtained from [88]. The $\mathbf{H}^+$ is pre-computed from $\mathbf{H}$ and the F blocks are partitioned according to their respective dimensions.

-2 Finite time ϵ-differential privacy

Theorem 1. *[87]: Let $q : D \to \mathbb{R}^k$ be a query, and $\epsilon > 0$. Then the Laplace mechanism $\mathcal{M}_q : D \times \omega \to \mathbb{R}^k$ defined by $\mathcal{M}_q(d) = q(D) + \eta$, with $\eta \sim Lap(b)^k$ and $b \geq \frac{\Delta(q)}{\epsilon}$ is ϵ-differentially private.*

Note that the mechanism requires each coordinate of to have standard deviation proportional to $\Delta_1 q$, as well as inversely proportional to the privacy parameter ϵ. For example, if q simply consists of k repetitions of the same scalar query, then $\Delta_1 q$ increases linearly with k, and the quadratically growing variance of the noise added to each coordinate prevents an adversary from averaging out the noise.

Proof. We have $R \subset \mathbb{R}^k$ measurable and d, d' two adjacent datasets in $\mathbb{D}$,

$$\begin{aligned} \Pr(\mathcal{M}_q(d) \in R) &= (\frac{1}{2b})^k \int_{\mathbb{R}^k} 1_R\,(q(d)+\eta) e^{-\frac{\|\eta|_1}{b}} d\eta \\ &= (\frac{1}{2b})^k \int_{\mathbb{R}^k} 1_R\,(s) e^{-\frac{\|s-q(d)|_1}{b}} d\eta \\ &\leq e^{\frac{\|q(d)-q(d')\|_1}{b}} (\frac{1}{2b})^k \int_{\mathbb{R}^k} 1_R(s)\, e^{-\frac{\|s-q(D)|_1}{b}} d\eta \end{aligned} \tag{1}$$

Since by triangular inequality, $-\|s-q(d)|_1 \leq \|s-q(d')|_1 + \|q(d)-q(d')|_1$, with $b = \Delta(q)/\epsilon$, we obtain the definition 10 of differential privacy.

□

-3 Minimum model error achieved by differential privacy

Definition 12. *Two state vectors $x_k, x'_k \in \mathbb{R}^n$ are said to be $\beta-$adjacent if there exists some $i = \{1, 2, \ldots n\}$ such that:*

$$\begin{aligned} \mathrm{Adj}^{\beta}(x) \overset{\text{def}}{=} |x_{i,k} - {x'}_{i,k}| &= \beta \qquad \forall \beta \geq 0 \\ x_{j,k} = {x'}_{j,k} &\quad \forall j \neq i \end{aligned}$$

Adj^{β} (x) implies that the state signals x and x' differ exactly by one component signal and that this deviation is bounded. In other words, differential privacy aims at hiding l_2 variations of size β in the signal x_i and makes it hard to detect deviations within that range.

Lemma 1. *The 1-norm of the uncertainty matrix Δ_A introduced in the model through the β-adjacent state vectors is norm (lower) bounded by*

$$\|\Delta_A\|_1 \;\geq\; \frac{|\,1 - \|A\|_1\,|}{1 + \frac{1}{\beta}\;\left\|\boldsymbol{X}_{[0:T-1]}\right\|_1}$$

Proof. To start with, we define the following relation:

$$\begin{aligned} A' = A + \Delta_A \text{ where } \Delta_A \text{ is the perturbation matrix} \\ \mathbf{X'}[0:T] = \mathbf{X}[0:T] + e[0:T] \quad \left\|e_{[0:T]}\right\|_1 \leq \beta \end{aligned} \tag{2}$$

$$\begin{aligned} x'_{k+1} = A'x'_k \quad = (A + \Delta_A)\, x'_k \\ x_{k+1} = Ax_k \end{aligned} \tag{3}$$

Looking at (2) and (3) we get:

$$\begin{aligned} \mathbf{X'}_{[1:T]} &= \mathbf{X}_{[1:T]} + e_{[1:T]} \\ A'\mathbf{X'}_{[0:T-1]} &= A\mathbf{X}_{[0:T-1]} + e_{[1:T]} \\ e_{[1:T]} &= (A + \Delta_A)\ \mathbf{X'}_{[0:T-1]} - A\, \mathbf{X}_{0:T-1} \\ \text{i.e. at any given k,}\ \ e_{k+1} &= A\, e_k + \Delta_A\, x'_k \end{aligned} \tag{4}$$

$$e_{[1:T]} = A\, e_{[0:T-1]} + \Delta_A\, \mathbf{X'}_{[0:T-1]} \tag{5}$$

Taking norm on both sides in (5) and applying the triangular inequality yields:

$$\left\|e_{[1:T]} - A\, e_{[0:T-1]}\right\| \leq \left\|\Delta_A\, \mathbf{X'}_{[0:T-1]}\right\| \tag{6}$$

Now, for some i we have the following:

$$e_{[1:T]} = \begin{bmatrix} 0 & 0 & \beta \cdots 0 \\ 0 & \beta & 0 \cdots 0 \\ \vdots & & \\ 0 & 0 & 0 \cdots \beta \end{bmatrix} \quad e_{[0:T-1]} \quad = \begin{bmatrix} 0 & \beta \cdots & 0 \\ \beta & 0 \cdots & 0 \\ \vdots & & \\ 0 & 0 \cdots & \beta \end{bmatrix} \tag{7}$$

$$\|e_{[1:T]}\|_1 = \|e_{[0:T-1]}\|_1 = \beta$$

$$A\, e_{[0:T-1]} = \begin{bmatrix} a_{11} & a_{12} \cdots a_{1n} \\ a_{21} & a_{22} \cdots a_{2n} \\ \vdots & \\ a_{n1} & a_{n2} \cdots a_{nn} \end{bmatrix} \begin{bmatrix} 0 & \beta \cdots & 0 \\ \beta & 0 \cdots & 0 \\ \vdots & & \\ 0 & 0 \cdots & \beta \end{bmatrix} = \begin{bmatrix} \beta a_{12} & \beta a_{11} & \cdots \beta a_{1n} \\ \beta a_{22} & \beta a_{21} & \cdots \beta a_{2n} \\ \vdots & & \\ \beta a_{n2} & \beta a_{n1} & \cdots \beta a_{nn} \end{bmatrix} \tag{8}$$

$$\begin{aligned} \left\|A\, e_{[0:T-1]}\right\|_1 &= \max\Big(\beta(|a_{12}| + |a_{22}| \cdots |a_{n2}|), \beta(|a_{11}| + |a_{21}| \cdots |a_{n1}|), \cdots \beta(|a_{1n}| + |a_{2n}| \cdots |a_{nn}|)\Big) \\ &= \beta\, \left\|A\right\|_1 \end{aligned}$$

Hence, the minimum value of $\left\|e_{[1:T]} - A\, e_{[0:T-1]}\right\| = \Big|\beta - \beta.|A\|_1\Big|$. Substituting this in (6), we arrive at

$$\begin{aligned} \left|\beta - \beta.|A\|_1\right| &\leq \|\Delta_A\|_1 \Big(\|\mathbf{X}_{[0:T-1]}\|_1 + \beta \Big) \\ \|\Delta_A\|_1 &\geq \frac{|\,1 - \|A\|_1\,|}{1 + \frac{1}{\beta}\ \left\|\mathbf{X}_{[0:T-1]}\right\|_1} \end{aligned} \tag{9}$$

□

Notice that as β increases the lower bound of $\|\Delta_A\|_1$ also increases. This matches with our intuition as the larger the potential state trajectory deviations get, the more ambiguous the generating model will become.

Glossary

Chapter 2

$\mathbb{R}$ -	Set of real numbers
T -	Time duration of a batch
x -	State vector
X -	State sequences up to time T
$\hat{X}$ -	Estimated state sequences up to time T
[.] -	Obfuscated or randomized value
$\mathbf{H}$ -	Grid topology/configuration or power network topology
$\mathbf{H}^+$ -	Pseudo-inverse of the grid configuration matrix
Pr -	Probability
L_i -	Locality or neighborhood i
S_{ij} -	Smart meter installed in household j in locality i
S_{i1} -	Lead smart meter at locality i
$S_{ij\ \forall j\neq 1}$ -	Other smart meter at locality i
Z_i -	Power consumption of locality i
m_i -	Number of smart meters in locality i
n_i -	Number of state variables in locality i
e -	Gaussian measurement noise
λ -	Key size in bits
$\mathcal{U}$ -	Utility provider
$\mathcal{C}$ -	Cloud service provider
$\mathcal{A}$ -	Area consisting of a set of localities
Δ -	100% dense matrix
$Obfuscate(.)$ -	Proposed protocol/solution/scheme

Chapter 3

DP -	Differential privacy
$w.r.t$ -	With respect to
$\mathbb{E}$ -	Expectation operator
x_k -	State vector at time instant k
$\tilde{x}_k$ -	Perturbed state vector at time instant k
$\mathbf{X}$ -	State sequences
$\tilde{\mathbf{X}}$ -	Perturbed state sequences
X_{avg} -	Average of the state vectors
$\tilde{X}_{avg}$ -	Perturbed average of the state vectors
η_k -	Noise vector at time instant k
n -	Number of state variables
T -	Time duration of a batch
$\|.\|_p$ -	p-norm of a vector
A -	System or state matrix
A' -	Adjacent state matrix
$\hat{A}$ -	Adversarial estimate
Q_k -	Query at a given instant k
$\Delta(x)$ -	Sensitivity function
$\Delta(T)$ -	Sensitivity up to time T
$\mathcal{U}$ -	Utility function
ϵ -	Privacy loss
Adj^{β} -	β adjacent
λ -	Privacy level
γ -	Eigenvalues
$\hat{\gamma}$ -	Adversarial estimate of eigenvalues
$\mathcal{M}$ -	DP Mechanism
R -	Range of the DP mechanism
$\mathbf{E}$ -	Estimation error