SECURITY AND PRIVACY OF CYBER-PHYSICAL SYSTEMS

by

WEIXIAN LIAO

Submitted in partial fulfillment of the requirements

For the degree of Doctor of Philosophy

Thesis Adviser: Dr. Pan Li

Department of Electrical Engineering and Computer Science

CASE WESTERN RESERVE UNIVERSITY

August, 2018

Security and Privacy of Cyber-Physical Systems

Case Western Reserve University
Case School of Graduate Studies

We hereby approve the thesis[1] of

WEIXIAN LIAO

for the degree of

Doctor of Philosophy

Dr. Pan Li

Committee Chair, Adviser Date
Department of Electrical Engineering and Computer Science

Dr. Kenneth Loparo

Committee Member Date
Department of Electrical Engineering and Computer Science

Dr. Michael Rabinovich

Committee Member Date
Department of Electrical Engineering and Computer Science

Dr. Mingguo Hong

Committee Member Date
Department of Electrical Engineering and Computer Science

04/20/2018

[1] We certify that written approval has been obtained for any proprietary material contained therein.

Dedicated to our NEST group at CWRU and my family.

Table of Contents

List of Figures

Acknowledgements

I would like to give my sincerest gratitude to Dr. Pan Li, my advisor, whose encouragement and insightful advices have been indispensable during my Ph.D. study journey. Most of the work presented herein comes from intellectual discussions I have with him over the last few years. As an outstanding researcher and advisor, Dr. Li have helped me to learn not only how to conduct research but also lifelong lessons.

I would also like to acknowledge my other committee members, Dr. Kenneth Loparo, Dr. Michael Rabinovich, and Dr. Mingguo Hong, for serving as my dissertation committee members and for their great help during my Ph.D. study.

I would like to extend my thanks to all my colleagues in our NEST group who provide me with encouraging research environment, their friendship and collaborations. I would like to specially acknowledge my colleagues and good friends, Sheng Cai, Xuhui Chen, Yifan Guo, Ming Li, Changqing Luo, Sergio Salinas, Arun Thapa, Qianlong Wang, Xufei Wang, Lixing Yu, and Kaijin Zhang, for countless valuable discussions and all the good times we have had.

Finally, I owe a special debt of gratitude to my family who have made enormous sacrifices for me and have always been there when I needed them the most. I owe all of my accomplishments to them.

Abstract

Security and Privacy of Cyber-Physical Systems

Abstract

by

WEIXIAN LIAO

Cyber-physical systems (CPS) are complex networked systems that consist of cyber components for computation and communication, closely interacting with physical components such as sensors and actuators. Recent years have witnessed exponential growth in the development of cyber-physical systems. As being the basis for emerging and future smart service, they play an increasingly important role in critical infrastructure, government, everyday lives, etc. On the other hand, the integration of CPS brings more threats that may result in catastrophic consequences for the society. In this dissertation, we aim to address the security and privacy issues in cyber-physical systems and internet of things (IoT) devices. Our contributions in this dissertation are two-fold. Firstly, we study the security issues in power grid, which is one of the most critical infrastructures in the world. Security of the power grid has gained enormous attention for decades. Cascading failure, one of the most serious problems in power systems, can result in catastrophic impacts such as massive blackouts. More importantly, it can be taken advantage by malicious attackers to launch physical or cyber attacks on the

power grid. However, due to the expansive geographical coverage and complex interdependencies among system components, protecting the power grid is data and computing intensive and hence extremely challenging. We investigate cascading failure attack (CFA) from a stochastic game perspective. In particular, we formulate a zero-sum stochastic attack/defense game for CFA while considering the attack/defense costs, limited budgets, diverse load shedding costs, and dynamic states in the system. Then, we develop a Q-CFA learning algorithm that works efficiently in a large system without any a-priori information. We also formally prove that the proposed algorithm can converge and achieve Nash equilibrium. Simulation results validate the efficacy and efficiency of the proposed scheme by comparisons with the state-of-the-art approaches.

Secondly, we focus on secure outsourcing of large-scale fundamental problems in the cloud. Conducting such large-scale data analytics in a timely manner requires a large amount of computing resources, which may not be available for individuals and small companies in practice. By outsourcing their computations to the cloud, clients can solve such problems in a cost-effective way. However, confidential data stored at the cloud is vulnerable to cyber attacks, and thus needs to be protected. Previous works employ cryptographic techniques like homomorphic encryption, which significantly increase the computational complexity of solving a large-scale problem at the cloud and is impractical for big data applications. We present an efficient secure outsourcing scheme for convex separable programming problems (CSPs). In particular, we first develop efficient matrix and vector transformation schemes only based on arithmetic operations that are computationally indistinguishable both in value and in structure under a chosen-plaintext attack (CPA). Then, we design a secure outsourcing scheme in which the client and the cloud collaboratively solve the transformed problems. The client can

efficiently verify the correctness of returned results to prevent any malicious behavior of the cloud. Theoretical correctness and privacy analysis together show that the proposed scheme obtains optimal results and that the cloud cannot learn private information from the client's concealed data. We conduct extensive simulations on Amazon Elastic Cloud Computing (EC2) platform and find that our proposed scheme provides significant time savings to the clients.

1 Introduction

1.1 Motivation

Cyber-physical systems (CPS) are complex networked systems that consist of cyber components for computation and communication, deeply intertwining with physical components such as sensors and actuators, on different spatial and temporal scales. Examples of cyber-physical systems are smart grid, autonomous systems, robotics systems, medical monitoring, automatic pilot avionics[1]. Recent years have witnessed exponential growth in the development of cyber-physical systems. As being the basis for emerging and future smart service, they play an increasingly important role in critical infrastructure, government, everyday lives, etc. For example, many wireless sensor networks monitor some aspects of environment and relay the collected and processed information to a central node. Another example is smart grid, which includes smart meters, smart appliances, renewable energy resources, etc., and provides more reliable, flexible, and efficient power services.

On the other hand, the integration of CPS brings more threats that may have catastrophic consequences for the society. For instance, security problem of the power gird has now been exaggerated due to various malicious cyber attacks that are launched on the power grid such as Denial-of-Service (DoS) attack[2], false data injection attack[3], energy theft attack[4], unobservable cyber attacks through topology errors[5], etc. Therefore, it motivates us to address the security and issues in cyber-physical systems and internet of things (IoT) devices.

In particular, we realize that cascading failure is a very concerning security problem in the power grid because some initial disturbances can trigger a series of unpredictable chain effects that possibly result in large-scale collapses in the system. This is exactly what happened in the 2003 Northeastern blackout, where the failure of a critical transmission line triggered a cascade of failure, resulting in shutting down the whole power system and affecting more than 55 million people in the Eastern U.S. and Canada[10]. Cascading failure has hence attracted intensive attention because of its criticality in the power grid. Chen et al.[7] propose a hidden failure model to assess the cascading dynamics in power systems. In[11], Rahnamay-Naeini et al. construct a probabilistic model for cascading failure while retaining key physical attributes and operating characteristics of power grids. Yan et al.[12] investigate the cascading failure by designing a new numerical metric called critical moment.

As cascading failure can lead to catastrophic damages in the power grid and can possibly take down the whole system, there is strong motivation for attackers to launch deliberate attacks by taking advantage of it, which we call "*cascading failure attacks (CFAs)*". However, analyzing CFA in the power grid is a very challenging problem because of the unpredictable cascading effect, the complex interactions between the attacker and the defender, the extremely high problem dimensionality in a large-scale system, etc.[14]. To the best of our knowledge, despite its importance, CFA has rarely been studied in the literature and hence deserves systematic investigation.

In this dissertation, we explore CFA in the power grid from a game theory perspective. Specifically, defending critical infrastructures against malicious attacks requires system operators to make optimal decisions about where to deploy limited budgets to improve the system resilience against adversaries. Game theory can be naturally employed to provide the system operators with such guidance on infrastructure protection[15–18]. For instance, Salmeron et al.[16] formulate the competition between a defender and an attacker as a leader-follower game. Chen et al.[17] propose a static game framework for defending the power system against deliberate attacks. Rao et al.[18] study a Stackelberg game while taking both the infrastructure survival probability and costs into account. These works consider the competition between the attacker and defender as an one-time event. However, power grid protection can be a continuous process where an attacker and a defender interact with each other many times at dynamic states[19]. For example, the nationwide power system in Yemen suffered from repeated attacks on transmission lines in 2014, which very soon left Yemen in total darkness[20]. Therefore, an attack-defense interaction model that considers dynamic system states and the long-term effects is indispensable.

To this end, we formulate a zero-sum stochastic game to characterize the long-term interactions between an attacker and a defender in CFA. Specifically, we consider that an attacker deploys limited budget to disrupt the components in the power grid, such as transmission lines, substations, etc. We consider that the attacker's objective is to maximize the total cost of the load shedding that is defined as a non-decreasing function of the total amount of shedding load, making the problem more challenging. On the other hand, a system defender deploys limited resources to minimize the total cost of load shedding by taking actions such as reinforcing a vulnerable transmission line or repairing a damaged line. Since the objectives of the attacker and the defender are opposite, we model the interactions between the two players as a zero-sum stochastic game. Stochastic games are difficult to solve due to the possible large problem dimensionality and their stochastic nature. Existing algorithms developed in the literature that are dynamic programming based algorithms, unfortunately, need to enumerate all the system states, the number of which is obviously too large in a large-scale power grid for the solution to be tractable. Thus, these algorithms suffer from the well known "curse of dimensionality" problem[22]. Furthermore, although such approaches are proven to converge to the optimum, they are under the assumption that all the dynamic system parameters, i.e., reward functions and transition probabilities, are always available for the players, which may not always be accessible in practice, especially to the attacker in the power grid. A couple of previous works on stochastic game analysis also assume complete a priori system information. Instead of having such strong assumptions, in this dissertation, we develop a Q-CFA learning algorithm to solve our stochastic game which can address the dimensionality problem and does not need any a priori system information. The intuition behind the learning process is that learning through past experience facilitates more intelligent decision makings and performance optimization.

Secondly, we focus on secure outsourcing of large-scale fundamental problems in the cloud. Conducting such large-scale data analytics in a timely manner requires a large amount of computing resources, which may not be available for individuals and small companies in practice. By outsourcing their computations to the cloud, clients can solve such problems in a cost-effective way. However, confidential data stored at the cloud is vulnerable to cyber attacks, and thus needs to be protected. Previous works employ cryptographic techniques like homomorphic encryption, which significantly increase the computational complexity of solving a large-scale problem at the cloud and is impractical for big data applications. On the other hand, we note that convex separable programming (CSP) is one of them that is involved in various real-world applications, including industrial control systems, time-dependent cost optimization, resource allocation, etc.[47–50]. For example, in the industry of water resource planning, sources that emit pollutants are required to remove waste from water system. However, solving CSPs is difficult[47,53], and becomes more challenging in big data. Specifically, large-scale CSPs are often too computationally complex to be solved by resource-limited users due to their limited computing capability and random access memory (RAM). To address this issue, many big companies and governments have to build supercomputer centers to conduct such heavy computation tasks. However, the expenditure is too high for individuals or small companies to afford. As a result, it is in dire need to find effective approaches to analyze large-scale data sets in a more efficient and economical way. Recently, researchers have suggested that cloud computing, which is characterized by robust computation power and pay-per-use manner, can be used to help resource-limited clients perform large-scale scientific computation and analytics[56–58]. In particular, clients can offload heavy computation tasks to the cloud and enjoy vast computation resources in a cost-effective manner. It has become widely utilized in various types of environments and supported clients to solve pressing issues in a more timely and cost-effective way. However, it also brings some serious concerns, one of which is data privacy. Clients' data often contains sensitive information, such as individuals' medical records, companies' proprietary information, engineering and scientific models, etc. The outsourcing paradigm of cloud computing deprives the clients' direct control on their private data, including both input and output privacy[60–63]. The leakage of such information may cause serious problems. For instance, in biomedical applications, a genomic database in the cloud is at risk of revealing the owners' DNA sequence; customers' shopping records in an e-commerce company may be stolen for unauthorized access to their behaviors; a grid company may suffer from cyber attacks if the system topology is disclosed[13,64]; and financial firms may be less competitive if their strategies are leaked. Therefore, in order to prevent the leakage of clients' private data, a good alternative is to allow clients to send their concealed data instead of real data to the cloud. Moreover, another issue is the verifiability of the results returned by the cloud. It is possible that the cloud may unintentionally or intentionally return invalid results. For example, if the software incurs some hardware failures or expensive cost during the operation, a malicious cloud may send incorrect results to the client. Consequently, a secure outsourcing protocol should be developed in a manner that enables the client to protect his/her data and check the correctness of the returned results as well. The last challenge is the computational efficiency. The additional burden incurred by the secure outsourcing scheme should be as little as possible. Otherwise there will be no incentive for the client to seek help from the cloud.

Therefore, the aforementioned challenges motivate us to design an efficient secure outsourcing scheme for convex separable programming problems (CSPs). In particular, we first develop efficient matrix and vector transformation schemes only based on arithmetic operations that are computationally indistinguishable both in value and in structure under a chosen-plaintext attack (CPA). Then, we design a secure outsourcing scheme in which the client and the cloud collaboratively solve the transformed problems. The client can efficiently verify the correctness of returned results to prevent any malicious behavior of the cloud. Theoretical correctness and privacy analysis together show that the proposed scheme obtains optimal results and that the cloud cannot learn private information from the client's concealed data. We conduct extensive simulations on Amazon Elastic Cloud Computing (EC2) platform and find that our proposed scheme provides significant time savings to the clients.

1.2 Scope and Organization of the Dissertation

The first goal of this dissertation is to design efficient secure outsourcing algorithm for large-scale convex separable programming problem in the cloud. Therefore, we discuss related work for security and privacy issues in cloud computing in Section 3.2. Section 3.3 introduces the system architecture, threat model, and security definitions. In section 3.4, we propose secure transformation and permutation algorithms to protect the original CSP problem with formal proofs. Section 3.5 presents an efficient transformation based scheme to solve the transformed large-scale CSP problem. The theoretical correctness and privacy analysis for the proposed schemes are discussed in Section 3.6. In Section 3.7, we evaluate the performance of the proposed algorithms through implementations on the Amazon Elastic Compute Cloud (EC2) platform and finally conclude this topic in section 3.8.

The second goal of this dissertation is to design an efficient algorithm for the formulated stochastic game and obtain the optimal attack and defense strategies for the attacker and defender respectively. To this end, we propose a Q-CFA algorithm and prove that the designed scheme achieves Nash Equilibrium. Specifically, in Chapter 2.2, we introduce our system models in detail, including DC power network model, cascading hidden-failure model, as well as the threat and defense models. In Chapter 3.3, we formulate the zero-sum stochastic game in the dynamic environment. In Chapter 2.4, we propose a Q-CFA learning algorithm to solve the formulated zero-sum stochastic game. We then prove that the proposed algorithm achieves the Nash Equilibrium. In Chapter 2.5, we present some simulation results to validate the efficacy and efficiency of our proposed algorithm. In Chapter 3.8, we conclude this topic and identify the another security and privacy problems in cyber-physical systems.

2 Cascading Failure Attacks in the Power System: A Stochastic Game Perspective

2.1 Introduction

The power grid is one of the most critical infrastructures in the world. Failure of the power grid can lead to severe economic, social, and security consequences, which makes security of the power gird a very crucial problem. This problem has now been exaggerated due to various malicious attacks that are launched on the power grid such as Denial-of-Service (DoS) attack[2], false data injection attack[3], energy theft attack[4], unobservable cyber attacks through topology errors[5], etc. On the other hand, due to its expansive geographical coverage and complex interdependencies among system components, protecting the power grid is data and computing intensive and hence extremely challenging[6].

Cascading failure is a very concerning security problem in the power grid because some initial disturbances can trigger a series of unpredictable chain effects that possibly result in large-scale collapses in the system. Taking the cascading failure in transmission networks[7,8] as an example, when one of the transmission lines fails and shifts its current load to the nearby lines, those connected lines may be pushed beyond their line capacities, become overloaded, and further shift their loads to other lines. Such sudden load spikes could induce the overloaded lines into failure, quickly spread the failure across other lines before the system operator can conduct any countermeasures, hence finally taking down the entire system in a very short time[9]. This is exactly what happened in the 2003 Northeastern blackout, where the failure of a critical transmission line triggered a cascade of failure, resulting in shutting down the whole power system and affecting more than 55 million people in the Eastern U.S. and Canada[10]. Cascading failure has hence attracted intensive attention because of its criticality in the power grid. Chen et al.[7] propose a hidden failure model to assess the cascading dynamics in power systems. In[11], Rahnamay-Naeini et al. construct a probabilistic model for cascading failure while retaining key physical attributes and operating characteristics of power grids. Yan et al.[12] investigate the cascading failure by designing a new numerical metric called critical moment.

As cascading failure can lead to catastrophic damages in the power grid and can possibly take down the whole system, there is strong motivation for attackers to launch deliberate attacks by taking advantage of it, which we call "*cascading failure attacks (CFAs)*". For example, a malicious attacker can launch a CFA to trip the critical transmission lines and in turn induce massive cascading failure[13]. However, analyzing CFA in the power grid is a very challenging problem because of the unpredictable cascading effect, the complex interactions between the attacker and the defender, the extremely high problem dimensionality in a large-scale system, etc.[14]. To the best of our knowledge, despite its importance, CFA has rarely been studied in the literature and hence deserves systematic investigation.

In this dissertation, we explore CFA in the power grid from a game theory perspective. Specifically, defending critical infrastructures against malicious attacks requires system operators to make optimal decisions about where to deploy limited budgets to improve the system resilience against adversaries. Game theory can be naturally employed to provide the system operators with such guidance on infrastructure protection[15–18]. For instance, Salmeron et al.[16] formulate the competition between a defender and an attacker as a leader-follower game. Chen et al.[17] propose a static game framework for defending the power system against deliberate attacks. Rao et al.[18] study a Stackelberg game while taking both the infrastructure survival probability and costs into account. These works consider the competition between the attacker and defender as an one-time event. However, power grid protection can be a continuous process where an attacker and a defender interact with each other many times at dynamic states[19]. For example, the nationwide power system in Yemen suffered from repeated attacks on transmission lines in 2014, which very soon left Yemen in total darkness[20]. Therefore, an attack-defense interaction model that considers dynamic system states and the long-term effects is indispensable.

To this end, we formulate a zero-sum stochastic game to characterize the long-term interactions between an attacker and a defender in CFA. Specifically, we consider that an attacker deploys limited budget to disrupt the components in the power grid, such as transmission lines, substations, etc. Maximizing the amount of load shedding due to disruption is usually adopted as the objective of the attacker in previous studies. However, loads on different transmission lines are of different importance to the system, and each transmission line contributes differently to the overall system reliability and security[7]. Therefore, we consider that the attacker's objective is to maximize the total cost of the load shedding that is defined as a non-decreasing function of the total amount of shedding load, making the problem more challenging. On the other hand, a system defender deploys limited resources to minimize the total cost of load shedding by taking actions such as reinforcing a vulnerable transmission line or repairing a damaged line. Since the objectives of the attacker and the defender are opposite, we model the interactions between the two players as a zero-sum stochastic game.

Stochastic games are difficult to solve due to the possible large problem dimensionality and their stochastic nature. Value iteration and policy iteration[21], i.e., iteratively improving the value functions or policies respectively, have been developed in the literature to solve this problem. Unfortunately, such dynamic programming based algorithms need to enumerate all the system states, the number of which is obviously too large in a large-scale power grid for the solution to be tractable. Thus, these algorithms suffer from the well known "curse of dimensionality" problem[22]. Furthermore, although such approaches are proven to converge to the optimum, they are under the assumption that all the dynamic system parameters, i.e., reward functions and transition probabilities, are always available for the players, which may not always be accessible in practice, especially to the attacker in the power grid. A couple of previous works on stochastic game analysis also assume complete a priori system information. Instead of having such strong assumptions, in this chapter, we develop a Q-CFA learning algorithm to solve our stochastic game which can address the dimensionality problem and does not need any a priori system information. The intuition behind the learning process is that learning through past experience facilitates more intelligent decision makings and performance optimization.

The goal of this work is to design an efficient algorithm for the formulated stochastic game and obtain the optimal attack and defense strategies for the attacker and defender respectively. To this end, we propose a Q-CFA algorithm and prove that the designed scheme achieves Nash Equilibrium.

In Chapter 2.2, we introduce our system models in detail, including DC power network model, cascading hidden-failure model, as well as the threat and defense models.

In Chapter 3.3, we formulate the zero-sum stochastic game in the dynamic environment.

In Chapter 2.4, we propose a Q-CFA learning algorithm to solve the formulated zero-sum stochastic game. We then prove that the proposed algorithm achieves the Nash Equilibrium.

In Chapter 2.5, we present some simulation results to validate the efficacy and efficiency of our proposed algorithm.

In Chapter 3.8, we conclude this chapter and identify the another security and privacy problems in cyber-physical systems.

2.2 System Models

In this chapter, we introduce DC power network model, cascading hidden failure model, as well as the threat and defense models used in our dissertation, respectively.

2.2.1 DC Power Network Model

We consider a power network consisting of $\mathscr{N} = \mathscr{G} \cup \mathscr{D}$ buses and $\mathscr{L} = \{1, \cdots, l, \cdots, L\}$ transmission lines. We assume that each bus is either a generation bus, denoted by $g \in \mathscr{G}$, or a load bus, denoted by $d \in \mathscr{D}$. Bus n_1 is identified as the reference bus. Similar to that in[23,24], we use DC power flow approximation of the AC system and assume that: 1) all bus voltage magnitudes are 1.0 per unit, 2) transmission line resistance is negligible, and 3) all bus voltage angles are small enough such that $sin(\theta_i - \theta_j) \approx \theta_i - \theta_j$, where θ_i and θ_j are the voltage angles at bus i and bus j respectively. Denote by $\Theta = [\theta_1, \cdots, \theta_n, \cdots, \theta_N]^T$, $\mathbf{P^G} = [p_1^G, \cdots, p_g^G, \cdots, p_G^G]^T$ and $\mathbf{D} = [d_1, \cdots, d_d, \cdots, d_D]$ as the bus voltage angle vector, the real power injection vector and the load demand vector respectively (note that $N = |\mathscr{N}|$, $G = |\mathscr{G}|$, and $D = |\mathscr{D}|$). Then, the DC power flow equations in matrix form, derived from the standard AC circuit equations and based upon the above assumptions[25], can be formulated as:

$$
\begin{aligned}
\mathbf{P^{inj}} &= \mathbf{K_p} \times \mathbf{P^G} - \mathbf{K_d} \times \mathbf{D}, && (2.1) \\
\Theta &= \mathbf{B} \times \mathbf{P^{inj}}, && (2.2) \\
f(l) &= b_{ij} \times (\theta_i - \theta_j), && (2.3)
\end{aligned}
$$

where $\mathbf{P^{inj}} = [p_2^{inj}, \cdots, p_n^{inj}, \cdots, p_N^{inj}]^T$ is the vector of nodal injection power for buses $2, ..., N$, $\mathbf{K_p}$ is the bus-unit incidence matrix, and $\mathbf{K_d}$ is the bus-load incidence matrix. θ_i and θ_j are the phase angles of bus i and bus j, respectively, that are connected by transmission line l. $f(l)$ is the real power flow on transmission line l. $\mathbf{B}$ is the $N \times N$ system susceptance matrix, in which $b_{ii} = \sum_{j \in S_i, j \neq i} \frac{1}{x_{ij}}$ and $b_{ij} = -\frac{1}{x_{ij}}$, where x_{ij} is the reactance between bus i and bus j. Notice that in this DC power network model, equation (2.1) is the power balance constraint, equation (2.2) calculates the phase angles for all the buses, which can be used for the power flow calculation on each transmission line in the network as shown in equation (2.3).

2.2.2 Cascading Hidden Failure Model

Hidden failure is among the top reasons for causing cascading failures in the power grid[7,11,12]. In this dissertation, we study the line protection hidden failure by considering the operation of protective relays, which are designed to trip the circuit breakers on the transmission lines when any fault is detected. Hidden failure is undetectable during the normal operation but will be exposed as a direct consequence of other system disturbances, for example, a sudden attack or natural disasters. Such sudden disturbances may cause the relay systems to inappropriately and incorrectly disconnect circuit elements. Thorp et al.[26] show that when transmission line l trips, because of the redistribution of the loads on it, hidden failures on all the lines connected with it will be exposed, i.e., those lines are then exposed to incorrect tripping probabilistically, because of the redistribution of the loads on the tripped line. Furthermore, if an exposed line trips, then the lines that are connected to this tripped line will be further exposed and subject to tripping probabilistically as well, which could eventually cause a cascade of failures and in the worst case, may spread the failure among the whole power grid and result in blackouts.

In this dissertation, we follow a general cascading hidden failure model[9,27]. Specifically, the probability for an exposed line to be tripped incorrectly is very low and considered as a constant p, when the load on this line is below its capacity, i.e., $F^{max}(l)$, and increases linearly to 1 when the load approaches $1.4 \times F^{max}(l)$. When the load on the line is or upon $1.4 \times F^{max}(l)$, this line will be tripped immediately for security purposes. This is consistent with the observed NERC events[28]. Thus, the probability of an exposed line tripping incorrectly, defined as $P_t(l)$, is

$$P_t(l) = \begin{cases} p, \text{ if } 0 \leq f(l) \leq F^{max}(l); \\ \frac{5(1-p)f(l)+7pF^{max}(l)-5F^{max}(l)}{2F^{max}(l)}, \\ \text{if } F^{max}(l) \leq f(l) \leq 1.4F^{max}(l); \\ 1, \text{ if } 1.4F^{max}(l) \leq f(l). \end{cases} \tag{2.4}$$

2.2.3 Threat Model

In the power grid, an attacker aims to disrupt the system by either physical attacks, e.g., severing transmission lines, damaging an critical associated transmission tower, or cyber attacks, e.g., false data injection attacks and DoS attacks[3]. The target of the attacks can be any components of the power system. Without loss of generality, in this dissertation, we consider the transmission lines as the attack targets, which are one type of the most common and far-ranging targets in the power system[29].

We first define two binary variables as follows:

$$\alpha(l) = \begin{cases} 1, \text{ if line } l \text{ is attacked;} \\ 0, \text{ otherwise.} \end{cases} \tag{2.5}$$

$$\delta(l) = \begin{cases} 1, \text{ if line } l \text{ is exposed;} \\ 0, \text{ otherwise.} \end{cases} \tag{2.6}$$

where $\alpha(l)$ is equal to 1 if the transmission line l is attacked by the attacker, and $\delta(l)$ is equal to 1 if line l is exposed according to the cascading hidden failure model in Chapter 2.2.2.

For practicality, we assume that the malicious attacker has limited budget to launch an attack. Specifically, it can only attack a limited number of transmission lines in one action. Therefore, the attacker's action is constrained by$\sum_{l\in\mathscr{L}} \mathbf{1}_{\alpha(l)=1} = b_a$, where b_a denotes the attacker's limited budget, i.e., the maximum number of transmission lines that it can attack in one action, and $\mathbf{1}_A$ is an indicator function that is equal to 1 when the event A is true and zero otherwise.

Subject to the budget constraint, the objective of the attacker is to cause the most damage to the power system. In the past, damage is simply measured as the total amount of loads that have to be shed due to the line failures[9]. However, since different loads may have different adverse impacts on the power system, it is more appropriate if we use the costs of load shedding as the objective of the attacker instead of the amount of load shedding. To this end, we denote the cost function on transmission line l as $u_l(\cdot)$, which is a nondecreasing function with regard to the shed load on the transmission line l, i.e., $\hat{d}(l)$. Consequently, the objective of the attacker is to maximize the total cost of load shedding in the power grid, i.e., to maximize $\mathscr{U} = \sum_{l\in\mathscr{L}} u_l(\hat{d}(l))$.

2.2.4 Defense Model

Similarly, a defender, who could be the power system operator or a third-party system protector, aims to protect the power grid from the attack. We define the available actions by the defender by reparing a damaged line or reinforcing an important line, i.e.,

$$\beta(l) = \begin{cases} 1, \text{if line } l \text{ is repaired or reinforced.} \\ 0, \text{otherwise.} \end{cases} \tag{2.7}$$

where $\beta(l)$ indicates if the defender chooses to repair the transmission line l or reinforce it.

We also assume that the defender has limited budget to protect the power grid, i.e., $\sum_{l\in\mathscr{L}} \mathbf{1}_{\beta(l)=1} = b_d$, where b_d denotes the defender's limited budget, i.e., the maximum number of transmission lines that it can repair or reinforce in one action.

Besides, the objective of the defender of the power grid is to find the best strategy that minimizes the total cost of load shedding in the power system, i.e., to minimize $\mathscr{U} = \sum_{l\in\mathscr{L}} u_l(\hat{d}(l))$.

Therefore, as the objectives of the defender and the attacker are opposite and the two players compete with each other at dynamic system states, we formulate a zero-sum stochastic game which will be introduced in the next chapter.

2.3 A Zero-sum Stochastic Game for CFA

As presented above, the objective of the attacker and that of the defender in CFA are opposite to each other. Therefore, in this chapter, we formulate a zero-sum stochastic game for the attacker and the defender in the power grid.

Before delving into the details of the formulation for the zero-sum stochastic game, we first briefly introduce stochastic games. In game theory, a stochastic game is a dynamic game with probabilistic transitions played by several players[30], which can be considered as an extension of Markov Decision Process[31]. The game is played in a sequence of stages. Specifically, at the beginning of each stage, the game is in some state. Players select actions independently and simultaneously based on their own budgets at the current state, and each player will receive an immediate reward that results from the chosen actions and the current state. Thereafter, this game moves to a new random stage whose transition probability is determined by both the actions from the players and the previous state. The procedure repeats continuously for a number of stages and each player endeavors to maximize their long-term reward, which is defined as the discounted sum of the immediate rewards at all stages.

2.3.1 States, Actions, and State Transitions

By considering the interactive competition between the attacker and the defender, we now formulate the CFA as a stochastic game **G**. In this game **G**, there are a set of system states, denoted by $\mathscr{S}$, in which each state $s \in \mathscr{S}$ is a vector that denotes the current status of all the transmission lines. Without loss of generality, we define the status of each transmission line as "up", denoted by u, or "down", denoted by w, when the line is functioning well or malfunctioning after being attacked, respectively. The stochastic game proceeds in a time-slotted fashion. Specifically, in each time slot, each player will choose an action based on the current system state so as to optimize its own objective. We denote by $\mathscr{M}_A(s)$ and $\mathscr{M}_D(s)$ the set of all the possible actions that the attacker and the defender can take, respectively, at state s. As discussed in Chapter 3.3.2 and Chapter 2.2.4, for the attacker, each $a \in \mathscr{M}_A(s)$ indicates the set of transmission lines to be attacked. On the other hand, for the defender, each $d \in \mathscr{M}_D(s)$ refers to a set of transmission lines to be repaired (if not working) or reinforced (if still working but vulnerable to attacks). Each action $a \in \mathscr{M}_A(s)$ and $d \in \mathscr{M}_D(s)$ will be selected by the attacker and the defender in each state s, respectively, with a certain probability denoted by $\pi_a(s)$ and $\pi_d(s)$.

Recall that each player selects their own actions independently and simultaneously in each stage. We denote p_{uwr} and p_{uw} as the probabilities for a functioning transmission line to fail upon attack with and without reinforcement by the defender in the same time slot, respectively. Similarly, we denote p_{wua} and p_{wu} as the probabilities for a non-functioning line to recover upon repair with and without being attacked in the same time slot, respectively. Obviously we have $0 \le p_{uwr} < p_{uw} \le 1$ and $0 \le p_{wua} < p_{wu} \le 1$. We can easily see that these probabilities can determine the transition probability $T(a, d, s, s')$ from state s to state s' under the actions a and d by the attacker and the defender, respectively. For example, suppose at the very beginning all lines in the system are up and there are no actions from the attacker or the defender. Then, when the attacker and the defender choose the same line to attack and reinforce respectively, the probability for the power system to remain in the same state is $1 - p_{uwr}$. Similarly, when the attacker attacks a line l and the defender chooses to reinforce another line l', the probability for the system to move to another state where only line l is down is p_{uw}.

2.3.2 Immediate Rewards

As mentioned before, the objectives of the attacker and the defender are opposite, i.e., maximizing/minimizing the total cost of the load shedding in the power grid. At each stage of the game, both players, i.e., the attacker and the defender, will receive immediate reward after taking actions. We define that with the actions by the attacker and the defender being a and d at state s, the immediate reward for the attacker, denoted by $U_A(a, d, s)$, is the total cost for load shedding.

We show in Fig. 2.1 what happens sequentially in one stage of the game where the attacker and the defender take actions a and d, respectively, at state s. Particularly, after both players take actions, some transmission lines might be tripped, and hence the system immediately adjusts according to the power equations (2.1)-(2.3)[32]. Then the system checks whether there are any lines overloaded. If so, the protective relays trip the overloaded lines and the system re-adjusts accordingly until there are no overloaded lines. Otherwise, the exposed lines, which share the same bus with the tripped lines, are

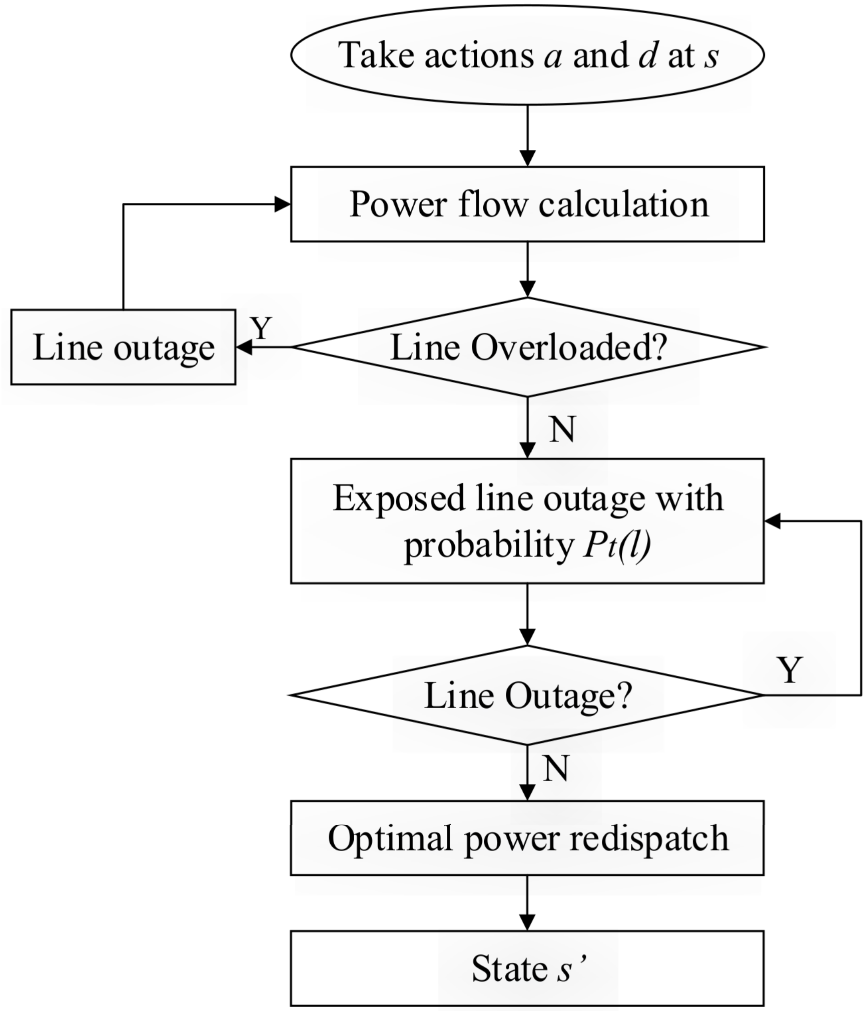

Figure 2.1. Flow chart for the power system after being attack

tripped with probability $P_t(l)$, based on the cascading model in Chapter 2.2.2. The cascading effect continues until there is no line outage any more. Finally, the power system performs security constrained optimal power flow redispatch, which is formulated as an optimization problem to minimize the total cost of load shedding, i.e., $\mathscr{U}$, in the current configuration of power system:

$$
\begin{aligned}
&\textbf{Minimize} \quad \mathscr{U}(a,d,s) = \sum_{l\in\mathscr{L}} u_l(\hat{d}_l), \\
&\textbf{s.t.} \quad \sum_{g\in\mathscr{G}} P_g + \sum_{l\in\mathscr{L}} \hat{d}_l - \sum_{l\in\mathscr{L}} d_l = 0 && (2.8)\\
&\qquad P_g^{min} \le P_g \le P_g^{max}, \quad \forall g \in \mathscr{G} && (2.9)\\
&\qquad -F^{min}(l) \le f(l) \le F^{max}(l), \quad \forall l \in \mathscr{L} && (2.10)\\
&\qquad 0 \le \hat{d}_l \le d_l, \quad \forall l \in \mathscr{L} && (2.11)
\end{aligned}
$$

where (2.8) is the power balance constraint, (2.9) is the generation capacity constraint for each generation unit, (2.10) limits the maximum power flow on each transmission line, and (2.11) indicates that the shed load cannot exceed the original load on the load bus.

Therefore, we have that the immediate reward for the attacker and that for the defender, known as the payoff of the game at state s are $\mathscr{U}(a,d,s)$ for all $a \in \mathscr{M}_A(s), d \in \mathscr{M}_D(s)$. Since the objective function is convex and all the constraints are linear, this problem can be easily solved and we can obtain the immediate rewards for each player at any system status.

Note that actions a and d executed at state s will bring the system state to the next possible state, resulting in further immediate rewards, i.e., $\mathscr{U}(a',d',s')$, at new state s'. Thus, actions taken at dynamic states will finally accrue a long-term reward as the game goes on. Both players' objectives are to obtain the optimal expected long-term rewards, which will be discussed next.

2.4 Optimal Strategies of the Stochastic Game

In this chapter, we first present the definition of optimal strategies. Then, we develop a Q-CFA learning algorithm to find the optimal strategies, which can work efficiently in large-scale systems.

2.4.1 Optimal Strategies

We refer to the optimal strategies as the mixed strategies of all actions chosen by the players that maximize their expected long-term rewards[33]. In this dissertation, we consider the case of stationary policies where action selection probabilities, i.e., $\pi_A(s)$'s and $\pi_D(s)$'s, do not change over time. In other words, we are interested in finding the convergent policies for each player at each state s.

From the attacker's point of view, we denote $V_A(s)$ as the attacker's expected long-term reward under the optimal strategies when the game starts at state s, and $Q_A(a,d,s)$ as the expected long-term reward for taking action a while the defender selects the action d when the game starts at state s. Specifically, we have

$$V_A(s) = \max_{\pi_A(s)} \min_{\pi_D(s)} \sum_{a\in\mathcal{M}_A(s)} \sum_{d\in\mathcal{M}_D(s)} \pi_a(s) Q_A(a,d,s) \pi_d(s), \tag{2.12}$$

where $\pi_A(s) = \{\pi_a(s) | a \in \mathcal{M}_A(s)\}$, $\pi_D(s) = \{\pi_d(s) | d \in \mathcal{M}_D(s)\}$, and

$$Q_A(a,d,s) = \mathcal{U}(a,d,s) + \gamma \cdot \sum_{s'\in\mathcal{S}} V_A(s') \cdot T(a,d,s,s'). \tag{2.13}$$

$V_A(s)$ and $Q_A(a,d,s)$ are also called the value of the state $s \in \mathcal{S}$ and the quality of the state s given actions a and d, respectively, for the attacker. $T(a,d,s,s')$ is the state transition probability from state s to state s' after taking actions a and d. Here the maximin function can be interpreted as follows. Since our game is a fully competitive stochastic game where each player selects an action independently and simultaneously at each system state, we need opponent-independent algorithms to solve this problem[34]. The maximin function makes (2.12) opponent-independent in which the attacker attempts to maximize its own expected long-term reward under the worst case assumption that the defender will always endeavor to minimize the payoff. Besides, note that (2.13) states that $Q_A(a,d,s)$ is equal to the immediate reward plus the discounted expected optimal value attainable from the next state s'. In (2.13), $\gamma \in [0,1)$ is a discount factor that represents how much impact the current decisions can have on the long-term reward. Particularly, when γ equals 0, the game becomes a one-time-event game[16–18]. When γ is larger than 0, a smaller value of γ emphasizes more the immediate rewards and a larger γ gives higher weight to the future rewards.

Similarly, the defender's expected long-term reward under the optimal strategies when the game starts at state s, denoted by $V_D(s)$, is

$$V_D(s) = \min_{\pi_D(s)} \max_{\pi_A(s)} \sum_{a \in \mathcal{M}_A(s)} \sum_{d \in \mathcal{M}_D(s)} \pi_a(s) Q_D(a,d,s) \pi_d(s), \tag{2.14}$$

where $Q_D(a,d,s)$ is the expected long-term reward for taking action d while the attacker selects the action a, i.e., the quality of the state s for the defender, and is formulated as

$$Q_D(a,d,s) = \mathscr{U}(a,d,s) + \gamma \cdot \sum_{s' \in \mathscr{S}} V_D(s') \cdot T(a,d,s,s'). \tag{2.15}$$

We note that in general $V_A(s) \leq V_D(s)$ due to weak duality, where $V_A(s)$ and $V_D(s)$ correspond to the primal problem and the dual problem, respectively. However, in a zero-sum stochastic game, strong duality holds and we have $V_A(s) = V_D(s) = V(s)$ (Section 5.4.5 in[35]). Consequently, the optimal solutions computed individually by the two players, i.e., $\pi^*_{\mathbf{A}}(s)$ and $\pi^*_{\mathbf{D}}(s)$, are the best responses to each other. We denote by $\pi^*(s) = \{\pi^*_{\mathbf{A}}(s), \pi^*_{\mathbf{D}}(s)\}$ the optimal strategy pair[36], which is known as the Nash equilibrium point in a stochastic game and defined as follows.

Definition 1. Nash Equilibrium: In a zero-sum stochastic game **G**, the Nash equilibrium for any state $s \in \mathscr{S}$ is an optimal strategy pair $\pi^*(s) = \{\pi^*_{\mathbf{A}}(s), \pi^*_{\mathbf{D}}(s)\}$ satisfying

$$V^{\pi^*(s)}(s) \geq V^{\{\pi_{\mathbf{A}}(s),\pi^*_{\mathbf{D}}(s)\}}(s),$$
$$V^{\pi^*(s)}(s) \leq V^{\{\pi^*_{\mathbf{A}}(s),\pi_{\mathbf{D}}(s)\}}(s).$$

Therefore, by finding the Nash equilibrium for each state s, we can obtain the attacker's and the defender's optimal strategies, i.e., essentially probability mass distributions on their action sets $\mathscr{M}_A(s)$ and $\mathscr{M}_D(s)$, which results in the optimal expected long-term reward for the attacker and the defender, respectively.

From the attacker's perspective, the optimal strategies $\pi^*_A(s)$ $(s \in \mathscr{S})$ can be found by solving (2.12) through algorithms like "value iteration"[21]. Particularly, at the k^{th} iteration, for each $s \in \mathscr{S}$, the attacker needs to solve the following optimization problem:

$$\begin{aligned}
&V^k_A(s) = \max_{\{\pi_a(s)\}} \min_{d \in \mathscr{M}_D(s)} \sum_{a \in \mathscr{M}_A(s)} Q^k_A(a,d,s) \cdot \pi_a(s) \\
\textbf{s.t. } &Q^k_A(a,d,s) = Q_c(a,d,s) + \gamma \cdot \sum_{s' \in \mathscr{S}} V^{k-1}_A(s') \cdot T(a,d,s,s') \\
&\sum_{a \in \mathscr{M}_A(s)} Q^k_A(a,d,s) \geq V^{k-1}_A(s) \\
&\sum_{a \in \mathscr{M}_A(s)} \pi_a(s) = 1 \\
&\pi_a(s) \geq 0, \forall a \in \mathscr{M}_A(s)
\end{aligned}$$

where $V^k_A(s)$ is the value of the state s in the k^{th} iteration. The basic idea of value iteration is that it iteratively estimates the value of $Q_A(a,d,s)$ and $V_A(s)$ using equations (2.12) and (2.13) for each $s \in \mathscr{S}$ in each iteration until convergence. The optimal strategies can then be obtained after scanning all the available states and action spaces. The defender can find its optimal strategies $\pi^*_{\mathbf{D}}(s)$ $(s \in \mathscr{S})$ by following a similar approach, which is omitted here due to space limit.

Value iteration has been proved to converge to the optimal results in stochastic games[37]. However, it assumes that the system information, such as the state transition probabilities $T(a,d,s,s')$'s, is a priori knowledge for both players, which may be inaccessible in practice. Moreover, this algorithm needs to enumerate all the system states and available actions in each iteration in order to obtain the optimal strategies. Nevertheless, the number of states and actions grow exponentially with the number of transmission lines, which obviously makes such algorithms fail to work in large-scale systems.

2.4.2 A Q-CFA Learning Algorithm

In order to account for the drawbacks of previous algorithms, we develop a machine learning based method, i.e., a Q-CFA learning algorithm based on the minimax-Q learning framework[33]. The proposed algorithm can gradually learn the optimal strategies without having any a priori knowledge of system information such as the state transition probabilities, i.e., $T(a,d,s,s')$'s. Besides, unlike value iteration and other previous algorithms, it does not need to scan all the states and actions in each iteration, and hence is scalable in large-scale systems.

The main idea of the proposed algorithm is as follows. Different from that in (2.13), we rewrite the quality of state s for the attacker under actions a and d by the attacker and the defender, respectively, i.e., $Q_A(a,d,s)$, at the k^{th} iteration into:

$$Q_A^k(a,d,s) = (1-\alpha(k)) \cdot Q_A^{k-1}(a,d,s) + \alpha(k) \cdot [\mathscr{U}(a,d,s) + \gamma V_A^{k-1}(s')] \tag{2.16}$$

where $\alpha(k) = \frac{1}{k+1}$ is the learning rate that decays over time, and s' is the next state after actions are executed in current state s. In other words, $Q_A^k(a,d,s)$ is updated by mixing the previous Q-value with a correction from the new estimate at a learning rate $\alpha(k)$. Then, the value of state s at the k^{th} iteration, i.e., $V_a^k(s)$, can be updated accordingly by

(2.12). Note that the quality and the value of state s for the defender can be updated in the same fashion.

Specifically, because of their limited budgets, both the attacker and the defender only have a limited number of actions at each stage of the game, which could be very diverse at different states. At the beginning of each state s_k, the algorithm firstly checks whether the current state has been observed in previous stages. If so, then both players use the previous profiles at state s_k to initialize the parameters such as the action sets, Q and V values. Otherwise, the algorithm initializes all the variables, and then adds the current state s_k into the observation history set denoted by H_s which contains profiles at all the past states. Subsequently, each player chooses an action. In particular, with a probability of p_{exp}, the attacker and the defender choose to explore their available action spaces, i.e., $\mathscr{M}_A(s)$ and $\mathscr{M}_d(s)$, respectively, and uniformly and randomly selects an action. This process is called exploration. On the other hand, with a probability of $1 - p_{exp}$, they choose to take the same actions selected in the previous initialization step, which is called exploitation. The intuition here is that the players in Q-learning can either randomly try out one of the available action profiles to possibly achieve higher reward in the long run, namely exploration, or attempt to maximize the reward by choosing the best known action, namely exploitation[38]. After both players take actions, they obtain their immediate rewards, update their Q and V function values, policies $\pi^*_A(s_k)$ and $\pi^*_D(s_k)$, and the learning rates $\alpha(k)$, respectively, and update the profiles for state s_k in the observation history set H_s. Thereafter, the game transits to the next state s_{k+1}. This procedure goes on until all states' policies have converged. The details of the proposed Q-CFA learning algorithm is described in Algorithm 1.

Notice that in order to update the profiles for each state, i.e., $(\pi_A^*(s_k), \pi_D^*(s_k))$, $V_A(s_k)$ and $V_D(s_k)$, in the Algorithm 1, we need to solve the subproblem of $\max_{\pi_A(s_k)} \min_{\pi_D(s_k)} \sum_{a\in\mathcal{M}_A(s_k)} \sum_{d\in\mathcal{M}_D(s_k)} \pi_a(s) Q_A^k(a,d,s_k)\pi_d(s_k)$ in the learning process, which turns out to be a matrix game where the strategies of attacker and defender form the row and column of the matrix respectively whose payoffs are $Q_A^k(a,d,s)$ and $Q_D^k(a,d,s)$ and we have that $Q_A^k(a,d,s) = Q_D^k(a,d,s_k) = Q^k(a,d,s_k)$. Therefore, we formulate the matrix game as:

$$\max_{\pi_A(s_k)} \min_{\pi_D(s_k)} \sum_{a\in\mathcal{M}_A(s_k)} \sum_{d\in\mathcal{M}_D(s_k)} \pi_a(s_k) Q^k(a,d,s_k)\pi_d(s_k) \tag{2.17}$$

However, the above optimization problem cannot be solved directly. In order to achieve the optimal strategies, i.e., $(\pi_A^*(s_k), \pi_D^*(s_k))$, we firstly assume that the attacker's strategies are fixed. Then the problem is reduced to:

$$\min_{\pi_D(s_k)} \sum_{a\in\mathcal{M}_A(s_k)} \pi_a(s_k) Q^k(a,d,s_k) \sum_{d\in\mathcal{M}_D(s_k)} \pi_d(s_k) \tag{2.18}$$

As $\sum_{a\in\mathcal{M}_A(s_k)} \pi_a(s_k) Q^k(a,d,s_k)$ is a vector, the solution to problem (2.18) is equivalent to searching for the smallest element in the vector, i.e., $\min_i [\sum_{a\in\mathcal{M}_A(s_k)} \pi_a(s_k) Q^k(a,d,s_k)]_i$. Thereafter, the matrix game (2.17) can be reformulated as:

$$\max_{\pi_A(s_k)} \min_i [\sum_{a\in\mathcal{M}_A(s_k)} \pi_a(s_k) Q^k(a,d,s_k)]_i \tag{2.19}$$

Next, we define $x = \min_i [\sum_{a\in\mathcal{M}_A(s_k)} \pi_a(s_k) Q^k(a,d,s_k)]_i$ and we have that $\left[\pi_a(s_k)Q^k(a,d,s_k)\right]_i \geq x$. Therefore, problem (2.17) can be further rewritten as:

$$\max_{\pi_A(s_k)} \quad x$$

$$s.t. \quad [\sum_{a\in\mathcal{M}_A(s_k)} \pi_a(s_k) Q^k(a,d,s_k)]_i \geq x \tag{2.20}$$

$$\sum_{a\in\mathcal{M}_A(s_k)} \pi_a(s_k) = 1 \tag{2.21}$$

$$\pi_a(s_k) \geq 0, \forall a \in \mathcal{M}_A(s_k) \tag{2.22}$$

Finally, we can transform this to a linear programming problem by viewing x as another variable:

$$\max_{\pi'} \quad \mathbf{0}_{aug}^T \pi'$$

$$s.t. \quad Q'\pi' \leq \mathbf{0} \tag{2.23}$$

$$\sum_{a\in\mathcal{M}_A(s_k)} \pi_a(s_k) = 1 \tag{2.24}$$

$$\pi_a(s_k) \geq 0, \forall a \in \mathcal{M}_A(s_k) \tag{2.25}$$

where $\pi' = [\pi_{\mathbf{a}}(\mathbf{s_k}), x]^T$, $Q' = ([\mathbf{0} \quad \mathbf{1}] - [\mathbf{Q^k}(\mathbf{a},\mathbf{d},\mathbf{s_k}) \quad \mathbf{0}])$ and $\mathbf{0}_{aug}^T = [\mathbf{0}^T \quad 1]$. Now since problem (2.23) is a linear progam, we can optimally solve the matrix game. Furthermore, as in each iteration of Algorithm 1, we optimally solve the subproblem, our algorithm converges to the Nash Equilibrium of the game, which is proved in next chapter.

2.4.3 Proof of the Nash Equilibrium

In what follows, we prove that our proposed algorithm converges to the Nash Equilibrium in the formulated zero-sum stochastic game. The general idea is that, we firstly prove the convergence of our algorithm, then prove that the obtained result is the Nash Equilibrium of the game as defined in Chapter 2.4.1.

Algorithm 1 Q-CFA Learning Algorithm

```
1: At State $s_k$, $k = 0, 1, ...$
   If state $s_t$ has been observed in any previous iteration, i.e., $s_t \in H_s$
   initialize $\pi_a$, $\pi_d$, $Q$, $V$ with the recorded profiles in $H_s$
   Otherwise,
   generate action sets $\mathscr{M}_A(s_k)$ and $\mathscr{M}_D(s_k)$,
   initialize $Q(a,d,s_k) \leftarrow 1$, for all $a \in \mathscr{M}_A(s_k)$ and $d \in \mathscr{M}_D(s_k)$,
   initialize $\pi_A(s_k) \leftarrow \frac{1}{|\mathscr{M}_A(s_k)|}$ and $\pi_D(s) \leftarrow \frac{1}{|\mathscr{M}_D(s_k)|}$,
2: Choose an action pair $\{\pi_{\mathbf{a}}, \pi_{\mathbf{d}}\}$ at state $s_k$:
   With probability $p_{exp}$, uniformly and randomly select an action in the action sets;
   Otherwise, return the action pair $\{\pi_a, \pi_d\}$ obtained in the initialization;
3: Learn and Update:
   Update $Q_A^k(a,d,s_k)$ according to (2.16), and $Q_D^k(a,d,s_k)$ similarly
   Update the optimal strategies $\pi_A^*(s_k)$ and $\pi_D^*(s_k)$ by
   $\pi_A^*(s_k) \leftarrow \arg\max_{\pi_A(s)} \min_{\pi_D(s)} \sum_{a \in \mathscr{M}_A(s_k)} \sum_{d \in \mathscr{M}_D(s_k)} \pi_a(s_k) Q_A^k(a,d,s_k) \pi_d(s_k),$
   $\pi_D^*(s_k) \leftarrow \arg\min_{\pi_D(s_k)} \max_{\pi_A(s_k)} \sum_{a \in \mathscr{M}_A(s_k)} \sum_{d \in \mathscr{M}_D(s_k)} \pi_a(s_k) Q_D^k(a,d,s_k) \pi_d(s_k)$
   Update $V_A(s_k)$ and $V_D(s_k)$ according to (2.12) and (2.14),
   Update $\alpha(k+1) \leftarrow \frac{1}{k+1}$;
4: The system transits to the next state $s_{k+1}$;
5: If all states' policies have converged, stop; otherwise, go to step 1.
```

1.8

Before we prove the convergence of the proposed algorithm, we have the following assumptions and lemma[39]:

Assumption 1. Every state and action have been visited infinitely often.

Assumption 2. The learning rate, $\alpha(k)$, satisfies the following conditions:

(1) $1 < \alpha(k) < 1$;

(2) $\sum_{k=0}^{\infty}(\alpha(k))^2 < \infty$.

Lemma 1. **(Conditional Average Lemma)** Under Assumptions 1 and 2, the process $V(k+1) = (1-\alpha(k))V(k)+\alpha(k)\omega(k)$ converges to $\mathbb{E}(\omega|h(k),\alpha(k))$, where $h(k)$ is the history at time stamp k.

Then, we arrive at a theorem for the convergence of our algorithm.

Theorem 1. In the proposed Algorithm 1, for any state $s \in \mathscr{S}$, the attacker's and the defender's policies, i.e., $\pi_A(s)$ and $\pi_D(s)$, converge to the Nash equilibrium point.

PROOF. In Algorithm 1, we have that the decaying learning rate $\alpha(k)$ is equal to $\frac{1}{k+1}$. Therefore, we can see that $0 < \alpha(k) < 1$, and $\sum_{k=1}^{\infty}(\alpha(k))^2 = \sum_{k=1}^{\infty}(\frac{1}{k+1})^2 < \sum_{k=1}^{\infty}(\frac{1}{k+1}\frac{1}{k}) = \sum_{k=1}^{\infty}(\frac{1}{k} - \frac{1}{k+1}) < \infty$.

For the attacker, by substituting (2.16) into (2.12), we get that for any $s \in \mathscr{S}$,

$$\begin{aligned}
&V_A^k(s)\\
=&\max_{\pi_A(s)}\min_{\pi_D(s)} \sum_{a\in\mathscr{M}_A(s)} \sum_{d\in\mathscr{M}_D(s)} \pi_a(s)\cdot\big[(1-\alpha^k(s))\cdot\\
&Q_A^{k-1}(a,d,s)+\alpha^k(s)\cdot(Q_c(a,d,s)+\gamma V_A^{k-1}(s'))\big]\cdot\pi_d(s)\\
=&(1-\alpha^k(s))V_A^{k-1}(s)+\alpha^k(s)\max_{\pi_A(s)}\min_{\pi_D(s)} \sum_{a\in\mathscr{M}_A(s)} \sum_{d\in\mathscr{M}_D(s)}\\
&\pi_a(s)\big(Q_c(a,d,s)+\gamma V_A^{k-1}(s')\big)\pi_d(s).
\end{aligned}$$

Define a mapping function T^k as

$$T^k V_A^k(s) = \mathbb{E}_{s'}\Big[\max_{\pi_A(s)}\min_{\pi_D(s)} \sum_{a\in\mathscr{M}_A(s)} \sum_{d\in\mathscr{M}_D(s)} \pi_a(s)\cdot\big(Q_c(a,d,s)+\gamma V_A^{k-1}(s')\big)\pi_d(s)\Big].$$

According to the Conditional Average Lemma, we can know that as the iterations in Algorithm 1 continue, $V_A^k(s)$ converges to $T^k V_A^k(s)$.

Next, we show that $T^k V_A^k(s)$ converges to the optimal value. Specifically, we can rewrite $T^k V_A^k(s)$ into:

$$\begin{aligned}
T^k V_A^k(s)=&\max_{\pi_A(s)}\min_{\pi_D(s)} \sum_{a\in\mathscr{M}_A(s)} \sum_{d\in\mathscr{M}_D(s)} \pi_a(s)\cdot\sum_{s'\in\mathscr{S}} T(a,d,s,s')\big(Q_c(a,d,s)+\gamma V_A^{k-1}(s')\big)\pi_d(s)\\
=&\max_{\pi_A(s)}\min_{\pi_D(s)} \sum_{a\in\mathscr{M}_A(s)} \sum_{d\in\mathscr{M}_D(s)} \pi_a(s)\cdot\big(Q_c(a,d,s)+\gamma\sum_{s'\in\mathscr{S}} V_A^{k-1}(s')T(a,d,s,s')\big)\pi_d(s).
\end{aligned}$$

We define another mapping function Z^{k-1} as

$$Z^{k-1}V_A^{k-1}(s) = \pi_a(s)\big(Q_c(a,d,s)+\gamma\sum_{s'\in\mathscr{S}} V_A^{k-1}(s')T(a,d,s,s')\big)\pi_d(s).$$

Z^{k-1} has been proved to be a contraction mapping in[40]. Therefore, $T^k V_A^k(s)$ is a contraction mapping as well.

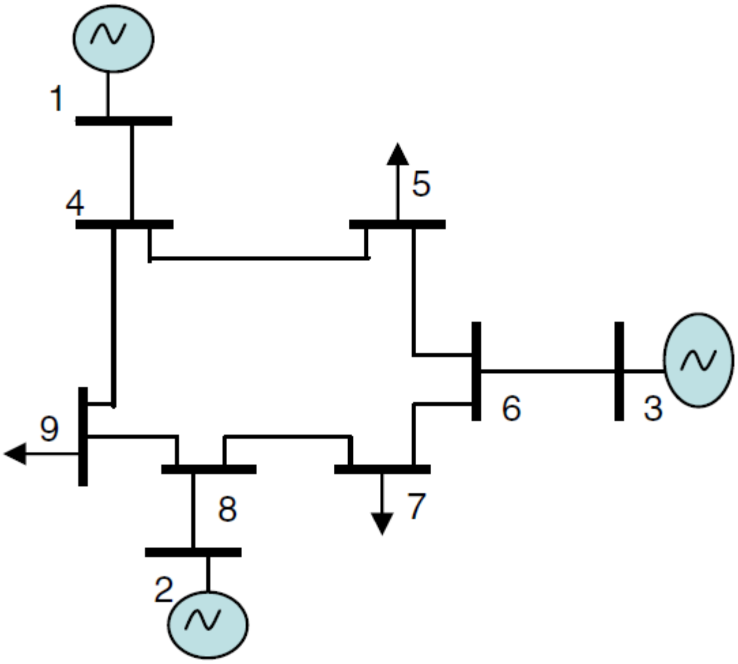

Figure 2.2. IEEE 9-bus system

Thus, we have

$$\begin{aligned} T^k(V_A^k)^*(s) = & \sum_{a\in\mathcal{M}_A(s)} \sum_{d\in\mathcal{M}_D(s)} \pi_a^*(s)\cdot\big(Q_c(a,d,s)+\gamma\sum_{s'\in\mathcal{S}} V_A^{k-1}(s')T(a,d,s,s')\big)\pi_d^*(s) \\ = & (V_A^k)^*(s), \end{aligned}$$

which means that $(V_A^k)^*(s)$ is the fixed point of T^k. According to Theorem 1 in[39], $V_A^k(s)$ converges to $(V_A^k)^*(s)$, i.e., $V^*(s)$, with probability 1.

Similarly, we can prove that $V_D^k(s)$ converges to $V^*(s)$ with probability 1 as well. Thus, this theorem directly follows. □

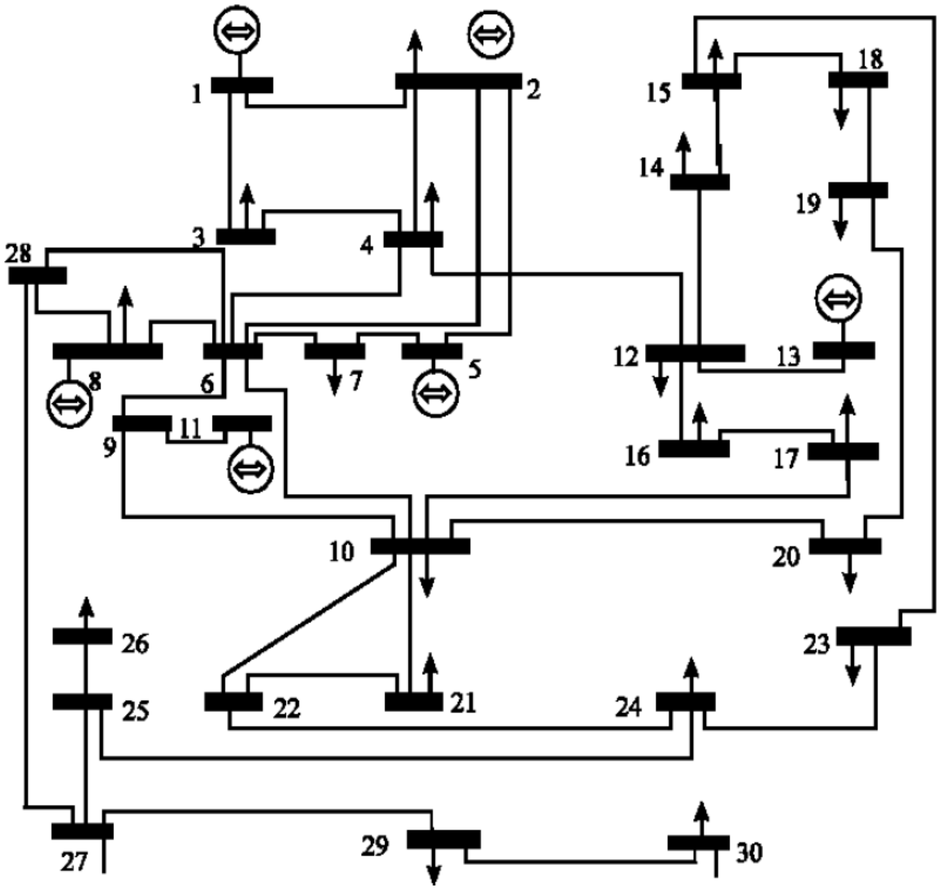

Figure 2.3. IEEE 30-bus system

2.5 Simulation Results

In this section, we conduct extensive simulations to demonstrate the efficacy and efficiency of the proposed scheme. We first demonstrate the convergence of our proposed Q-CFA algorithm in different systems. Then, we analyze the system operator's optimal strategies in different scenarios. Finally, we compare the system operator's expected long-term cost in our scheme with that in other existing schemes.

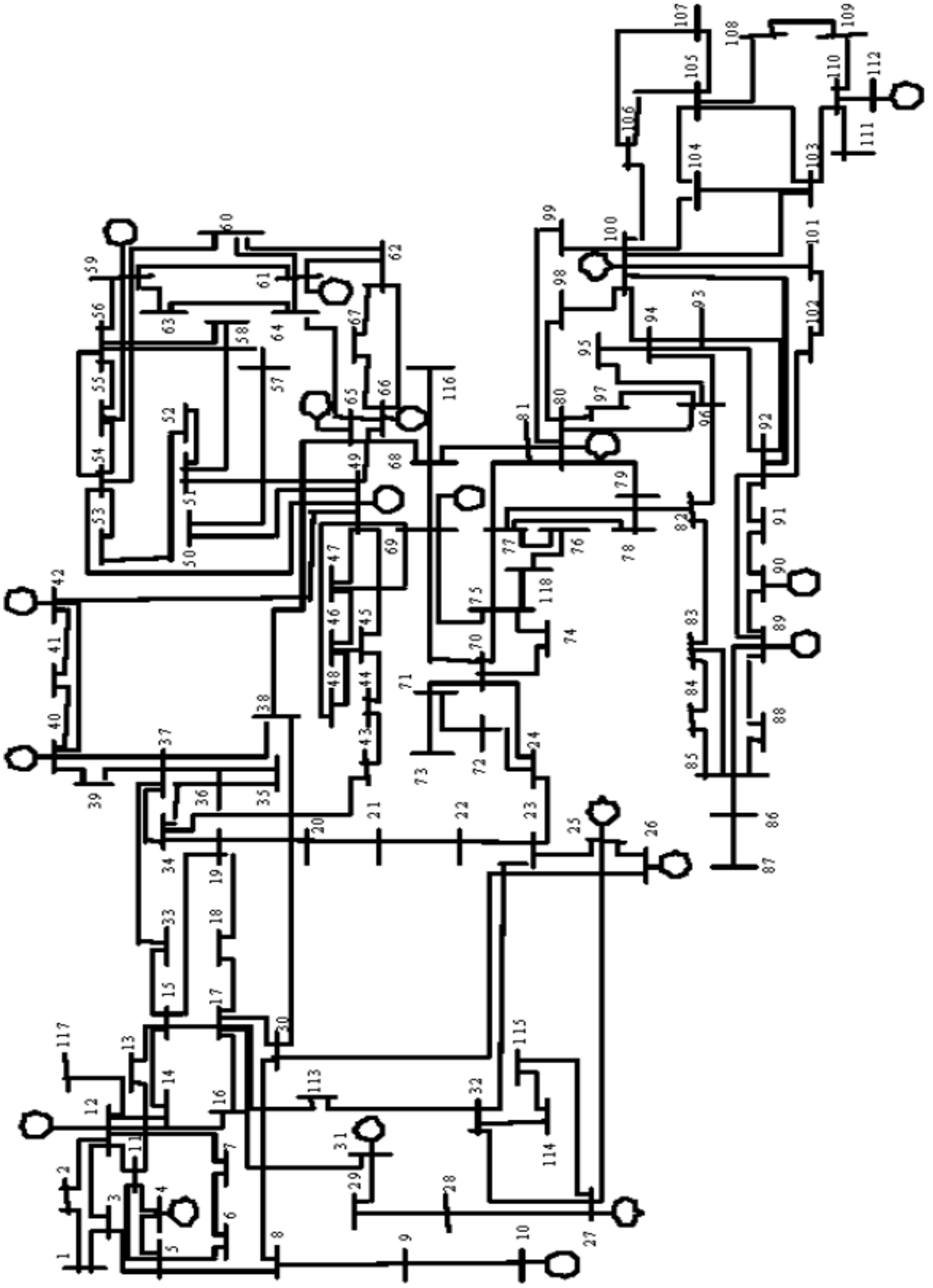

Figure 2.4. IEEE 118-bus system

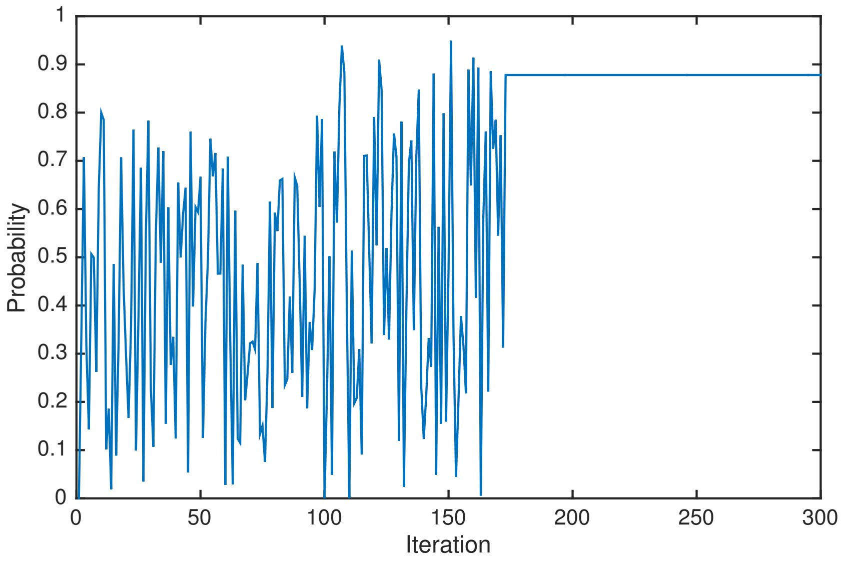

Figure 2.5. Attacker's strategy on line 7 at state 0 in the IEEE 9-bus system

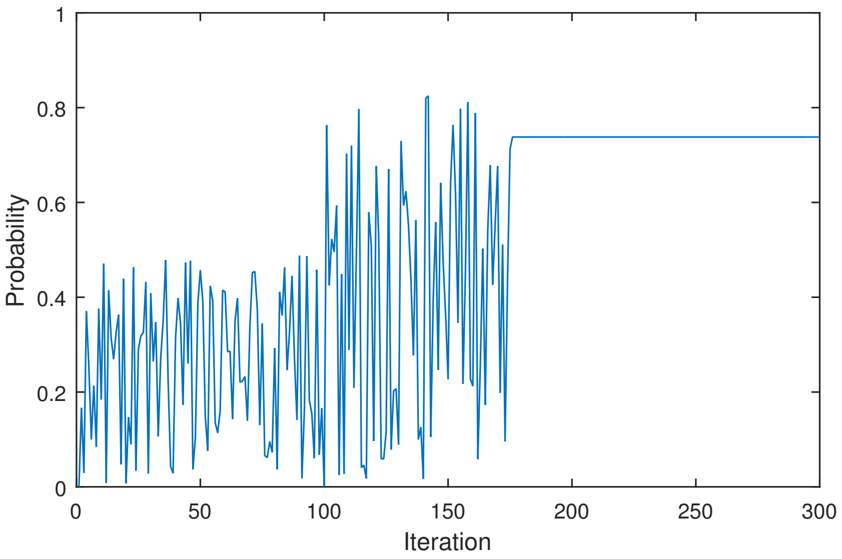

Figure 2.6. Defender's strategy on line 7 at state 0 in the IEEE 9-bus system

2.5.1 Convergence of Q-CFA

We first study the convergence of the proposed Q-CFA algorithm using the IEEE standard 9-bus, 30-bus and 118-bus systems, respectively, and the MATPOWER toolbox[41].

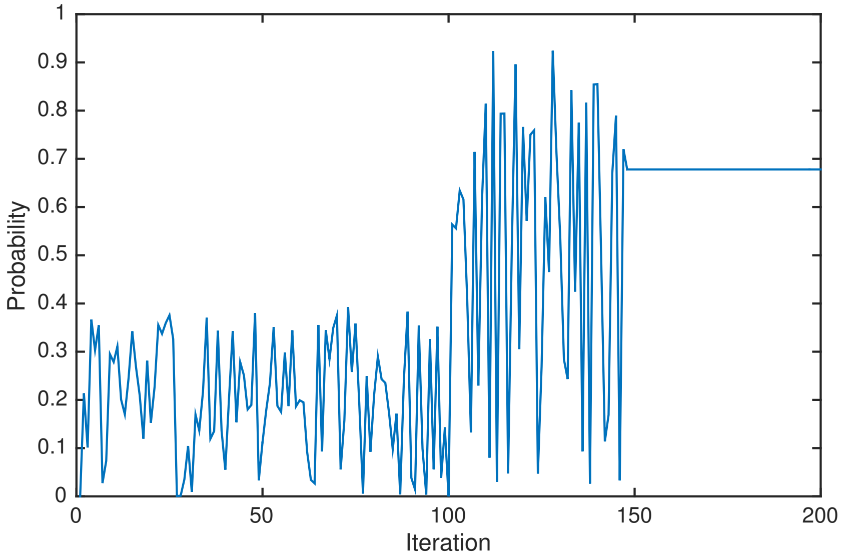

Figure 2.7. Attacker's strategy on line 3 at state in the IEEE 9-bus system7

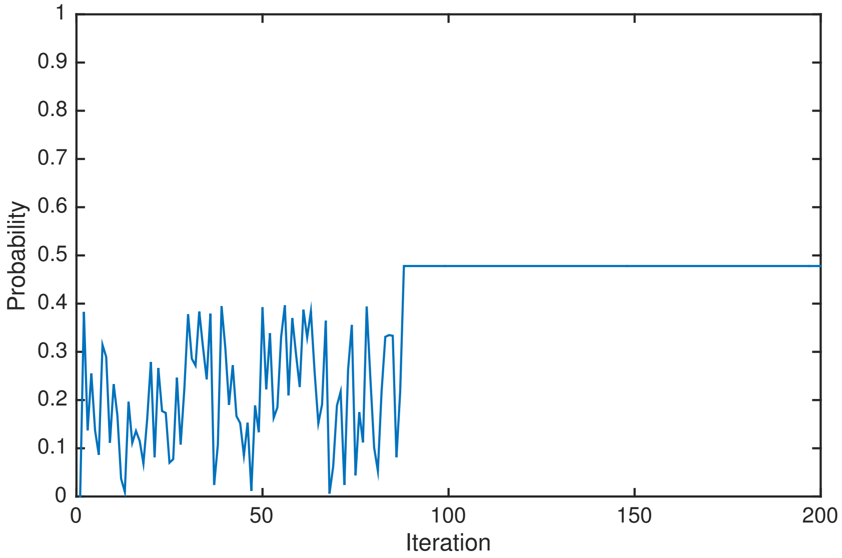

Figure 2.8. Defender's strategy on line 7 at state 7 in the IEEE 9-bus system

As IEEE 118 bus test system does not include flow limits, we employ the flow limits in Table 3 (the transmission line data) in[42]. In Fig. 2.2, Fig. 2.3, and Fig. 2.4, we show the configuration of standard IEEE bus systems used in our experiments. To initialize

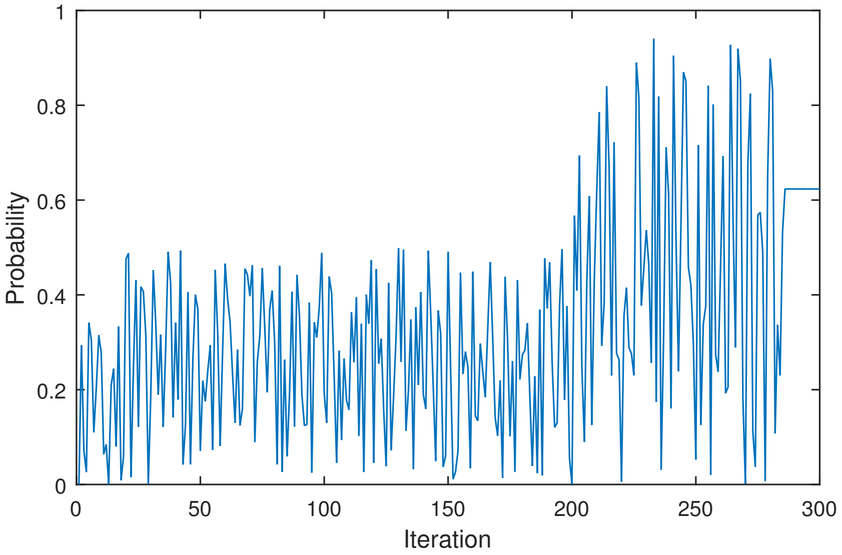

Figure 2.9. Defender's strategy on line 29 at state 0 in the IEEE 30-bus system

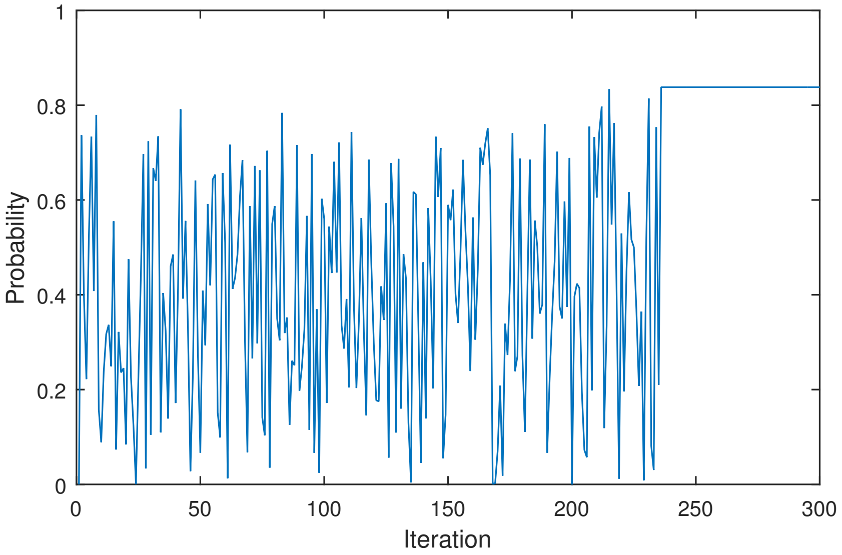

Figure 2.10. Attacker's strategy on line 7 at state 0 in the IEEE 30-bus system

the simulation, we set the transition probabilities $p_{uw} = 0.5$, $p_{uwr} = 0.3$, $p_{wu} = 0.5$, $p_{uwa} = 0.3$, the discounting factor $\gamma = 0.3$ and the exploration probability $p_{exp} = 0.6$. For illustrative purposes, we consider that the resources of each player are normalized

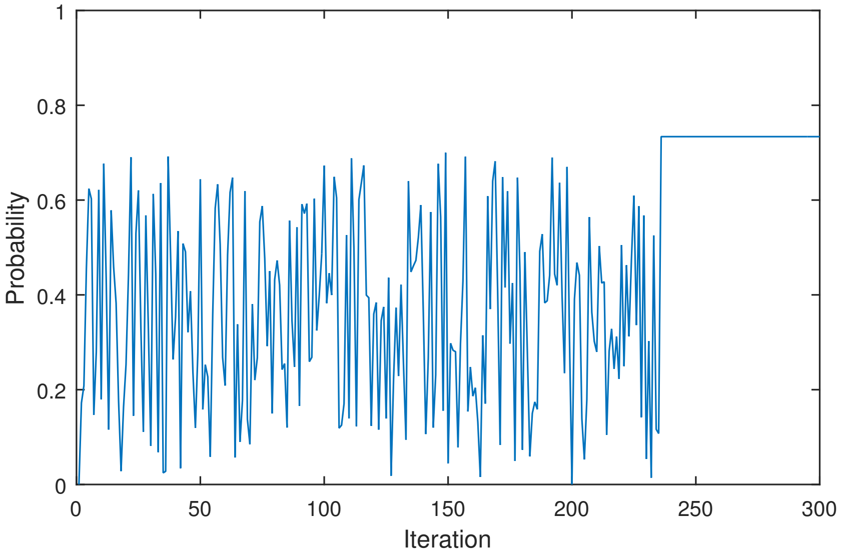

Figure 2.11. Attacker's strategy on line 16 at state 27 in the IEEE 30-bus system

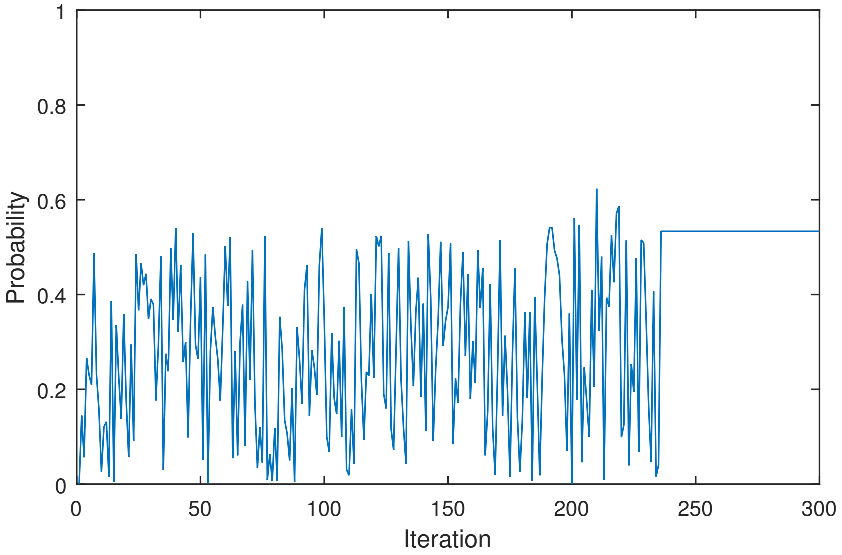

Figure 2.12. Defender's strategy on line 27 at state 27 in the IEEE 30-bus system

to one, particularly, each player can affect one transmission line in one time slot. Because each transmission line is of different importance to the entire system, we set different load shedding cost for each line. Specifically, we define the load shedding cost as a linear function of the amount of shed loads on line l and is given by

$$u_l(\hat{d}_l) = c_l \hat{d}_l, \tag{2.26}$$

where c_l is a given positive constant for line l. We conduct experiments on a desktop with a 3.41 GHz i7-6700 CPU, 16GB RAM and a 1TB hard disk drive. To demonstrate the convergence of our proposed Q-CFA, we show in Fig. 2.5 - Fig. 2.16 the learning curves of the system operator's and the attacker's strategies at certain states in the IEEE 9-bus, 30-bus, and 118-bus systems, respectively. For instance, line 3 and line 7 are the most important lines in the IEEE 9-bus system, which become the main targets in the players' optimal strategies as shown in Fig. 2.5 and Fig. 2.6. In particular, the attacker and the system operator tend to attack and defend, respectively, the transmission line 7 when the game starts. It indicates that when all the transmission lines are well functioning, the most critical line in the IEEE 9-bus system is the line 7. As the iteration goes by, both the attacker and defender's strategies converge and the obtained strategies are stationary, which means the mixed strategies do not change over time. When the state of the game transits to state 7 where line 7 is malfunctioning, as shown in Fig. 2.7 and Fig. 2.8, we can see that the system operator is more likely to repair line 7 but the attacker more likely turns to attack line 3. We can also observe similar results in the IEEE 30-bus and 118-bus systems. Noticeably, from Fig. 2.5 - Fig. 2.16, we can find that both players' strategies converge within 200, 250, 400 iterations in the IEEE 9-bus, 30-bus and 118-bus systems, respectively. Since we have proved that the converged strategies are the Nash equilibrium points, the results in the simulation are optimal under dynamic environments. Moreover, from a game-theoretic perspective, the strategies obtained by our proposed algorithm will serve as guidance for the system operator to deploy either

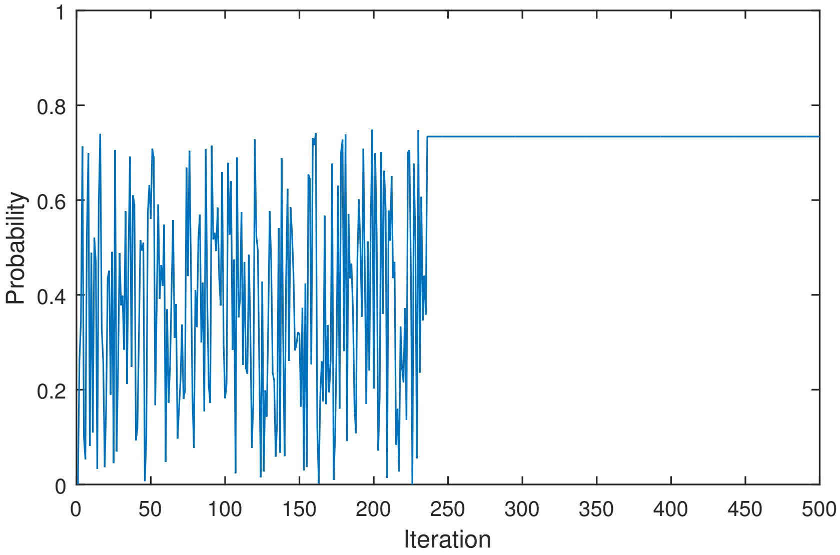

Figure 2.13. Attacker's strategy on line 9 at state 0 in the IEEE 118-bus system

reinforcement or repair on system components in different system configuration under the condition that the attacker targets the most critical system components. By doing so, the system operator can reduce the risk of having cascading failures, and hence the expected long-term costs.

2.6 Strategy Analysis

Next, we analyze the system operator's optimal strategies in the stochastic game when the discount factor γ varies, with γ being equal to 0, 0.3, 0.8. Recall that $\gamma \in [0,1)$ represents the impact that current decisions can have on the long-term reward. Particularly, when γ equals 0, the game becomes a static game. When γ is larger than 0, a smaller value of γ emphasizes more on the immediate rewards and a larger γ gives a higher weight to the future rewards. In Fig. 2.17, compared with the results in the static game where $\gamma = 0$, the performance in the stochastic games where $\gamma > 0$ is much better. This

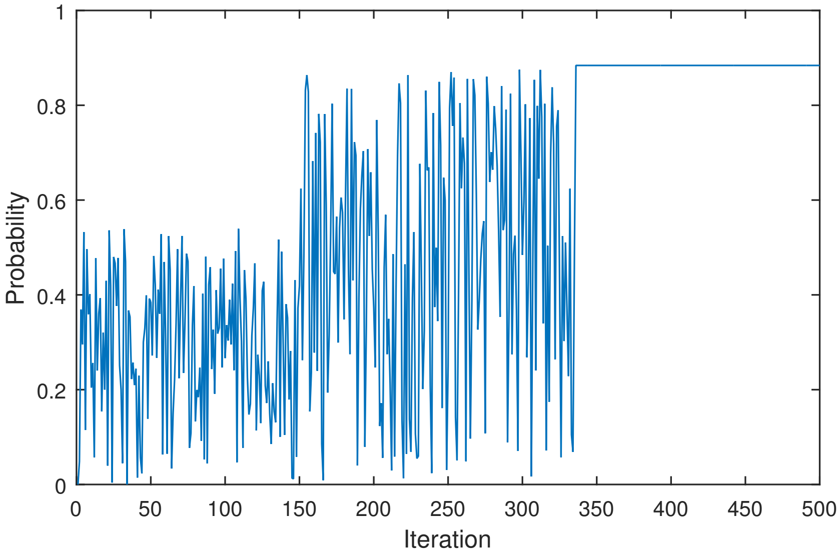

Figure 2.14. Defender's strategy on line 7 at state 0 in the IEEE 118-bus system

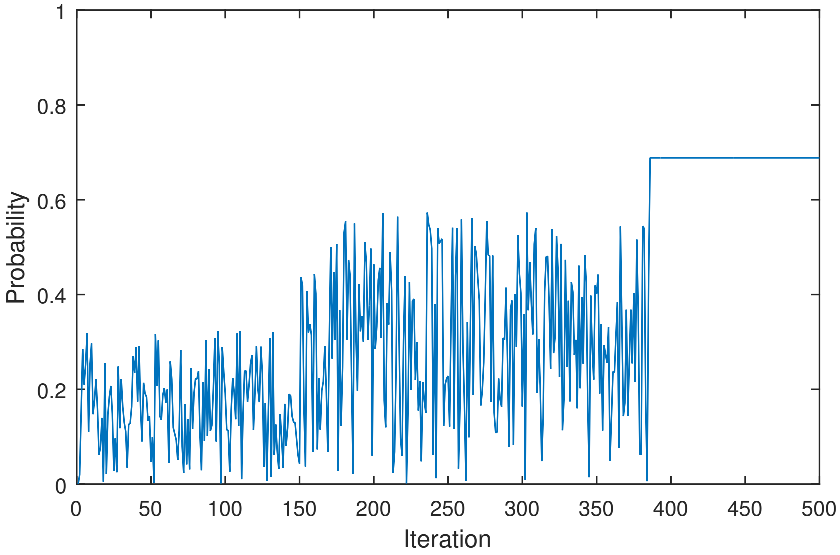

Figure 2.15. Attacker's strategy on line 8 at state 9 in the IEEE 118-bus system

is because in the stochastic games, players not only care about current rewards, but also take the future states into consideration. By considering both the current and future rewards, players are able to obtain optimal expected long-term rewards. In addition, we

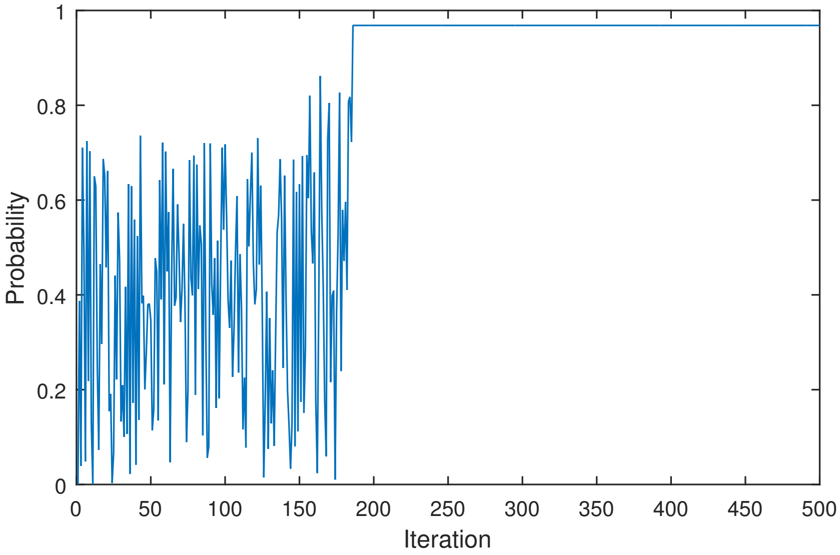

Figure 2.16. Defender's strategy on line 9 at state 9 in the IEEE 118-bus system

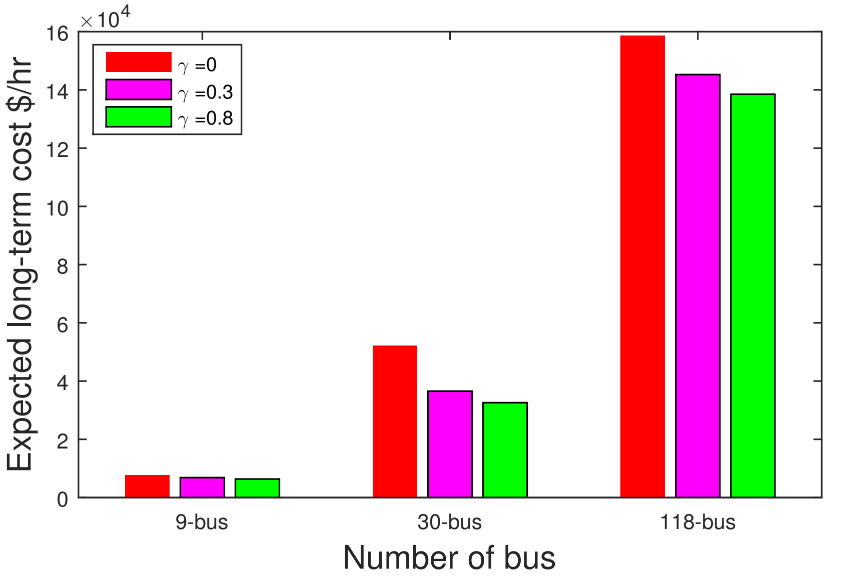

Figure 2.17. Performance analysis with regard to different γ

can see that the higher γ is, the lower expected long-term load shedding cost the system operator can achieve. This is because when γ increases, the system operator emphasizes

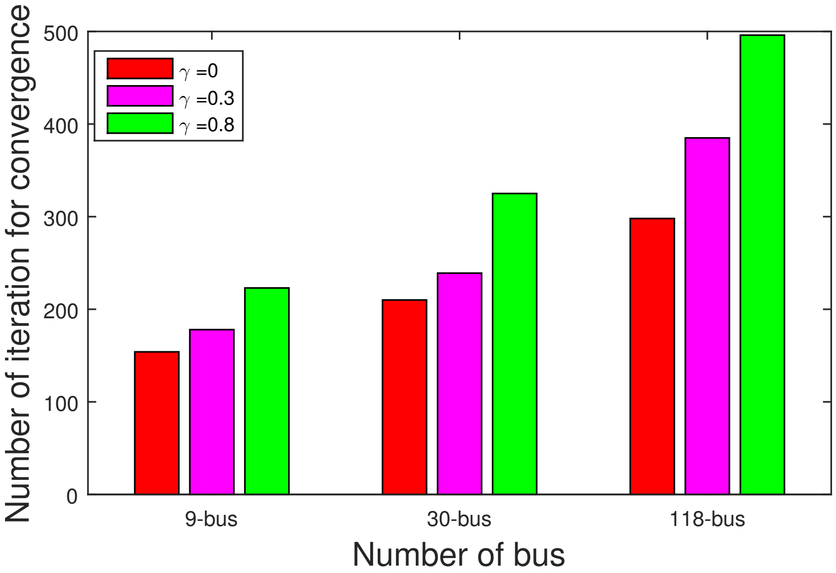

Figure 2.18. Convergence analysis with regard to different γ

more on the future states and can better react to the dynamic environments, which result in more savings in the long-term cost. On the other hand, Fig. 2.17 and Fig. 2.18 together demonstrate the tradeoff between performance and computational cost. As shown in Fig. 2.18, the number of iterations needed for convergence increases as γ increases. This is because when we emphasize more on the future rewards, it takes more iterations to search for the optimal solution.

2.6.1 Performance Comparison

Finally, from the system operator's perspective, we compare the performance of the optimal strategies obtained by our Q-CFA algorithm with that of two other strategies, i.e., the fixed strategy and the myopic learning strategy. In particular, in the fixed strategy, the system operator will draw an action o uniformly from the available action space, i.e., $\mathscr{M}_O(s)$, for each state s. In the myopic learning strategy where the game is a static game ($\gamma = 0$), the system operator only considers immediate rewards and ignores the

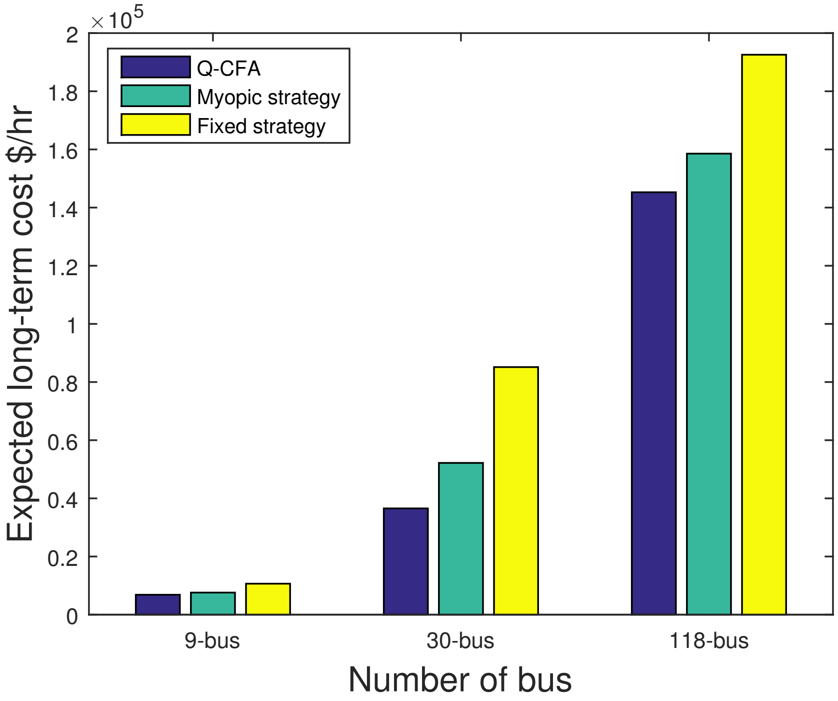

Figure 2.19. Performance comparison among three strategies.

impact of the current action on future rewards. Note that it is of paramount importance to select initiating events in each algorithm because it allows the attacker to determine if the initial event can cause a cascading failure. In the three benchmark algorithms, the selections of "important line" are different. In particular, our proposed scheme optimizes the expected long-term rewards, so the selection of initiating events takes the opponent's strategy and the dynamic environments into consideration. However, as the myopic strategy is a static-game strategy, selection of initiating event only considers the opponent's strategy in current state and the strategy can be explained as trying to launch a one-time attack to cause cascading failure and achieve the maximum immediate reward. On the other hand, the fixed strategy is a uniform strategy for comparison. So the selection of initial event is uniformly distributed. We compare the optimal expected long-term cost in these three strategies in Fig. 2.19.

We can find that the optimal costs obtained by our proposed Q-CFA and the myopic learning strategy are much lower than that obtained by the fixed strategy. This is because both of our proposed Q-CFA and the myopic learning strategy try to minimize the attacker's maximal reward, while the fixed strategy only uniformly chooses actions from the available action set without taking the opponent's possible strategies into consideration. In addition, because our Q-CFA algorithm optimizes the expected long-term reward while the myopic learning strategy only focuses on optimizing the strategies at the current state, our scheme outperforms the myopic learning strategy in the long run. Therefore, as a power system operator, adopting our proposed Q-CFA algorithm to defend the power system can both adapt to the dynamic state changes and attacker's intelligent strategies, which results in the best performance in the long run.

2.7 Conclusions

The IoT technologies have brought both new features and significant security challenges to power systems. In this dissertation, we have investigated CFAs in power systems. Specifically, we have formulated a zero-sum stochastic game to analyze the interactions between an attacker and a system operator in dynamic environments for power systems. This problem is very complex and computationally intensive. Different from the previous work where complete enumeration of the system states is required, making the algorithms computationally intractable for large-scale power system applications, we pro- pose an efficient Q-CFA learning algorithm that only searches certain related possible actions for each player in the game, making the scheme scalable with fast convergence. We have also theoretically proven that the proposed algorithm achieves the

Nash equilibrium. Moreover, considering that real-time statistics and sensitive data like system transition probabilities may not be accessible in practice, which unfortunately is an indispensable assumption in previous algorithms, our scheme works efficiently without requiring a priori knowledge of the system transition states. Simulation results show that by considering the system dynamics and the opponent's possible strategies, the optimal policy obtained by our proposed Q-CFA algorithm can achieve much better performance compared to several benchmark schemes.

3 Efficient Secure Outsourcing of Large-scale Convex Separable Programming

3.1 Introduction

The amount of data in our world has grown tremendously and double every two years, increasing from 4.4 zetabytes (million terabytes) in 2013 to 44 zetabytes by 2020[43]. The large-scale data sets, known as big data, have become a key basis of innovation and intelligence. They bring new opportunities to many areas such as scientific research, business innovations, human well-being, etc. For example, biomedical researchers develop personalized medicine programs to significantly improve patient care by finding patterns in large-scale genomic databases[44]; e-commerce companies, such as Amazon and eBay, provide accurate merchandise recommendations for customers by analyzing billions of transactions[45]; power system engineers perform real-time analysis and operations based on the massive amount of data collected from smart meters[46].

A critical underlying task of the aforementioned applications is to solve a series of large-scale fundamental problems. We note that convex separable programming (CSP) is one of them that is involved in various real-world applications, including industrial control systems, time-dependent cost optimization, resource allocation, etc.[47–50]. For example, in the industry of water resource planning, sources that emit pollutants are required to remove waste from water system. The decision makings can be formulated as CSPs where the pounds of biological oxygen demands are variables and the objective is to minimize total costs to the region while meeting specified pollution standards[51]. Another example is smart grid operations. Particularly, the objective function can be maximizing the revenue of a big company with regards to monthly energy consumptions at different sub-companies, while the total energy cost in each month is upper-bounded. Obviously, this problem can be formulated as a CSP as well. In addition, it is a useful mathematical tool as we can convert general nonlinear programming problems into CSP problems[52]. For instance, with the help of feedforward neural networks, general non-separable functions can be approximated as convex separable functions and the original problems can be transformed into CSP problems. Therefore, solutions to CSPs are very useful to many complex scientific and engineering problems.

However, solving CSPs is difficult[47,53], and becomes more challenging in big data. Specifically, large-scale CSPs are often too computationally complex to be solved by resource-limited users due to their limited computing capability and random access memory (RAM). To address this issue, many big companies and governments have to build supercomputer centers to conduct such heavy computation tasks. However, the expenditure is too high for individuals or small companies to afford. As a result, it is in dire need to find effective approaches to analyze large-scale data sets in a more efficient and economical way. Recently, researchers have suggested that cloud computing, which is characterized by robust computation power and pay-per-use manner, can be used to help resource-limited clients perform large-scale scientific computation and analytics[54–58]. In particular, clients can offload heavy computation tasks to the cloud and enjoy vast computation resources in a cost-effective manner. It has become widely utilized in various types of environments and supported clients to solve pressing issues in a more timely and cost-effective way. For example, financial corporations can outsource the intensive computation analysis of frequent stock deals to the cloud and obtain fast response to markets for realtime high frequency trading[59]. It is very impractical for the companies to run it with limited computational resources, which leads to delayed responses to markets and may cause inestimable losses. To give another example, smart grid companies can outsource complex power distribution schemes to the cloud for contingency analysis and power flow optimization, which can save noticeable computational resources, improve energy efficiency, and obtain realtime safety responses.

In spite of the enormous benefits, cloud computing also brings some serious concerns, one of which is data privacy. Clients' data often contains sensitive information, such as individuals' medical records, companies' proprietary information, engineering and scientific models, etc. The outsourcing paradigm of cloud computing deprives the clients' direct control on their private data, including both input and output privacy[60–63]. The leakage of such information may cause serious problems. For instance, in biomedical applications, a genomic database in the cloud is at risk of revealing the owners' DNA sequence; customers' shopping records in an e-commerce company may be stolen for unauthorized access to their behaviors; a grid company may suffer from cyber attacks if the system topology is disclosed[13,64]; and financial firms may be less competitive if their strategies are leaked. Therefore, in order to prevent the leakage of clients' private data, a good alternative is to allow clients to send their concealed data instead of real data to the cloud. Moreover, another issue is the verifiability of the results returned by the cloud. It is possible that the cloud may unintentionally or intentionally return invalid results. For example, if the software incurs some hardware failures or expensive cost during the operation, a malicious cloud may send incorrect results to the client. Consequently, a secure outsourcing protocol should be developed in a manner that enables the client to protect his/her data and check the correctness of the returned results as well. The last challenge is the computational efficiency. The additional burden incurred by the secure outsourcing scheme should be as little as possible. Otherwise there will be no incentive for the client to seek help from the cloud.

To account for these challenges, there have been a bunch of works studying privacy issues in cloud computing. However, the current literature overlooks the secure outsourcing schemes for large-scale CSPs. It is indeed challenging and different from before since clients only allow very few local computing and storage resources, which significantly limits the amount of computations that can be operated by themselves to preserve the privacy of the data. To tackle this challenge, we study the secure outsourcing of CSPs. To this end, we propose an efficient secure outsourcing algorithm for solving large-scale CSPs. Specifically, we consider a CSP where the objective function and constraints are composed of convex functions. Firstly we develop an efficient transformation scheme to preserve the privacy of vectors and matrices. We prove that the secure transformation of vector and matrices is computationally indistinguishable both in value and structure under a chosen plaintext attack (CPA). Then, we utilize the characteristics of CSPs and linearize the convex functions in the CSP problem with arbitrary accuracy, which results in solving a series of secure large-scale linear programming (LP) problems in the cloud. Next, we securely outsource the LP problems to the cloud for solutions. To ensure the returned results' integrity, we adopt a light-weight scheme to effectively verify the correctness of the final results. Our main contributions in this chapter are summarized as follows:

- We develop an efficient transformation based scheme to solve the secure outsourcing computation of large-scale CSPs. This is among the first studies in the literature to investigate this problem.
- Our secure outsourcing scheme is based on very efficient arithmetic operations instead of heavy computations like homomorphic encryptions.
- We show that the proposed secure transformation of vector and matrices is computationally indistinguishable both in value and in structure under a chosen-plaintext attack (CPA).
- Experimental results show that this proposed algorithm achieves noticeable time savings.

The rest of this chapter is organized as the follows. We discuss related work for privacy issues in cloud computing in Section 3.2. Section 3.3 introduces the system architecture, threat model, and security definitions. In section 3.4, we propose secure transformation and permutation algorithms to protect the original CSP problem with formal proofs. Section 3.5 presents an efficient transformation based scheme to solve the transformed large-scale CSP problem. The theoretical correctness and privacy analysis for the proposed schemes are discussed in Section 3.6. In Section 3.7, we evaluate the performance of the proposed algorithms through implementations on the Amazon Elastic Compute Cloud (EC2) platform and finally conclude this chapter in section 3.8.

3.2 Related Work

To tackle the data privacy issues in outsourcing paradigm for cloud computing, there have been some existing schemes that are designed and applied to encrypt and outsource basic mathematical problems. For example, Yan et al.[65] propose a deduplication scheme while preserving the privacy of data storage in the cloud. Jiang et al.[66] investigate the secure data integrity auditing for shared dynamic data. Our previous works in[67,68] study the secure outsourcing of linear systems of equations and quadratic programming problems. Zhou et al. provide a privacy-preserving outsourcing tool that focuses on a quadratic programming problem[69]. Outsourcing methods for modular exponentiation, image reconstruction, linear regression and database are also reported in[70–73], respectively. Moreover, there are privacy-preserving outsourcing schemes for matrix operations, including matrix inversion[60,74], matrix determinant[75], and matrix multiplication[76]. These works can be generally classified into two categories: cryptographic approaches and transformation based approaches. In particular, there are some works based on traditional cryptographic techniques for secure outsourcing of large-scale computations to the cloud to protect and analyze clients' data. Gennaro et al.[77] propose a fully homomorphic encryption (FHE) scheme that enables secure outsourcing of a function to the cloud. Wang et al.[59] develop an iterative algorithm in which the cloud and a client solve a linear system of equations collaboratively. However, protecting data privacy requires applying partial homomorphic encryption on the data by the client, which has a high computational complexity ($\mathcal{O}(\log_2 e)$ flops per encrypted value, were e is the key size). Similarly, Liu et al.[78] employ homomorphic encryption to solve gradient descent problems in the cloud. It still requires the client to perform computationally expensive operations to guarantee the theoretical privacy. On the other hand, by using homomorphic encryption, the client forces the cloud to carry out operations on ciphertexts, which increases formidable overhead to the already computationally expensive computations in the cloud since ciphertexts need to be handled with specialized linear algebra software.

Besides the cryptographic techniques, mapping functions are also being used in the literature for securely outsourcing problems to the cloud. For instance, Lei et al.[60] and Atallah et al.[79] design secure algorithms that use linear algebra operations to outsource the matrix inversion problem to the cloud. However, since matrix inversion is usually an intermediate step to solve other problems (e.g., in the solution of linear systems of equations, the coefficient matrix needs to be multiplied by the constant vector after inversion), it may incur heavy communication cost, and sometimes is even infeasible, to communicate the matrix back to the client before the algorithm can continue. Besides, Wang et al.[80 81] design a private outsourcing scheme for linear programming problems by applying an affine mapping function to the objective function and constraint matrices. However, it is prohibitively expensive for the client to carry out matrix-matrix multiplications and other cryptographic computations. We find that such works impose large computational complexity on the local client, which may not be practical for large-scale data sets. In our previous work[67], we propose to offload the heaviest computations of an iterative algorithm for solving the large-scale linear systems of equations whose coefficients have been randomized. The computational complexity is low, whereas the client needs to exchange vectors with the cloud at every iteration, which incurs communication delays.

3.3 Problem Formulation

In this section, we introduce our system architecture, the threat model, and security definitions, respectively.

3.3.1 System Architecture

We consider a two-party computing architecture for large-scale CSPs as shown in Fig. 3.1, where a client has a resource-limited computing device and a remote cloud server has abundant computing capabilities. The client tries to solve a large-scale CSP problem with the help of the cloud by outsourcing the most computationally complex tasks to the cloud to find the optimal solution while preserving his/her data privacy. A large-scale CSP problem can be formulated as follows:

$$\begin{aligned} \textbf{P1:} \quad & \textbf{Min} \quad F = \sum_{j=1}^{n} f_j(x_j), \\ \textbf{s.t.} \quad & \sum_{j=1}^{n} g_{ij}(x_j) \le b_i, i = 1, \cdots, m && (3.1) \\ & x_{jL} \le x_j \le x_{jU}, j = 1, \cdots, n && (3.2) \end{aligned}$$

where F is a nonlinear separable function. $f_j(x_j)$'s and $g_{ij}(x_j)$'s are general convex functions. b_i's are constants. x_{jL}'s and x_{jU}'s are lower and upper bounds for x_j's. Problem **P1** is said to be a convex separable programming (CSP) problem because all the variables, i.e., $x_j, j \in [1, n]$, are mathematically independent in the convex objective and convex constraint functions[82].

CSPs[47] are a special class of optimization problems, which arise frequently in practical applications such as time-dependent optimization in industry applications. For

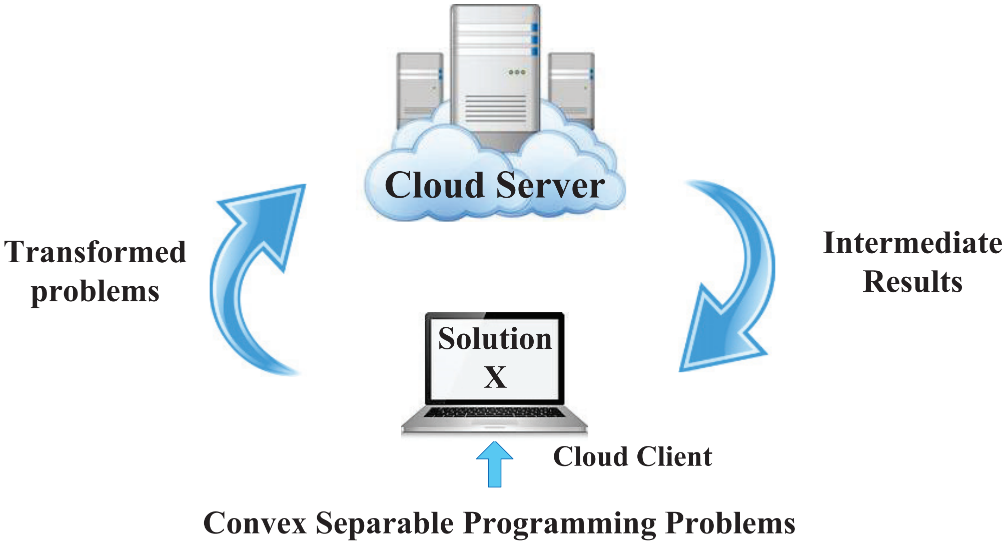

Figure 3.1. A secure architecture for outsourcing separable programming problem

example, in an industrial resource utilization problem, each variable x_j represents the resource utilization in the time period j, and the results of the resource utilization or profits are additive over time. Thus, this problem can be formulated as a CSP problem where decision variables are subject to practical constraint functions. Another example is smart grid operations. Particularly, the objective function can be minimizing zero minus the revenue of a big company with regards to monthly energy consumptions at different sub-companies, while the total energy cost at different months are upper-bounded. Obviously, this problem can be formulated as a CSP as well.

Besides, we denote the set of index pairs that point to non-zero elements in a general matrix $\mathbf{K} \in \mathbb{R}^{m \times n}$ as follows:

$$\mathscr{S}_{\mathbf{K}} = \{(i,j) | k_{i,j} \neq 0 \ \forall i \in [1,m], \forall j \in [1,n]\} \tag{3.3}$$

where i and j denote the ith and jth column of $\mathbf{K}$, respectively.

3.3.2 Threat Model

We consider that the cloud server is malicious and knows the proposed secure outsourcing algorithm for the CSP problem. Specifically, while the cloud follows the protocol and looks for solutions to the problem, it tries to extract knowledge from the client's data and the final results as well. The cloud may even try to deviate from the proposed protocols and return erroneous results so as to save computing resources or due to some hardware failures. We also consider that in the problem **P1**, the objective functions, $f_j(x_j)$'s, constraint functions $g_{ij}(x_j)$'s, the lower and upper bounds, i.e., x_{jL}'s and x_{jU}'s, all contain sensitive information that should not be revealed to the cloud. The optimal solution x_j's and the optimal value of the objective function should not be known by the cloud either.

3.3.3 Security Definition

In this study, we adopt the definition of computational indistinguishability based on the chosen-plaintext attack (CPA)[83] in our secure outsourcing scheme design. In particular, we regard that both the values and positions of non-zero elements in a matrix are private information. In what follows, we formally define computational indistinguishability under a CPA, known as CPA security, for two types of private information, respectively.

Definition 2. Pseudorandom Function Let Φ be a function and ϕ a truly random function. We say that Φ is a pseudorandom function if for all probabilistic polynomial-time distinguishers D, there exists a negligible function μ such that

$$|Pr[D^{\Phi}(1^n) = 1] - Pr[D^{\phi}(1^n) = 1]| \leq \mu \tag{3.4}$$

Distinguishers D^{Φ} and D^{ϕ} have oracle access to functions Φ and ϕ, respectively.

Definition 3. Computational Indistinguishability in Value We say that a matrix transformation scheme has indistinguishable transformations in value under a chosen-plaintext attack (or is CPA-secure in value) if for all probabilistic polynomial-time adversaries $\mathscr{A}$, there exists a negligible function μ, such that the probability of distinguishing two matrix transformations in value in a CPA indistinguishability experiment is less than $1/2+\mu$.

Definition 3 establishes the inability of an attacker to tell apart the non-zero values in a matrix **K** from those in another matrix.

Moreover, the positions of the non-zero elements in **K** (i.e., **K**'s structure), contain private information that should also be hidden from the cloud. To protect a matrix's structure, we propose to permute its rows and columns in such a way that the non-zero elements occupy positions that are indistinguishable from those of the non-zero elements in another matrix. We give the definition of secure permutation below.

Definition 4. Computational Indistinguishability in Structure We say that a permutation scheme has indistinguishable permutations under a chosen-plaintext attack (or is CPA-secure in structure) if for all probabilistic polynomial-time adversaries $\mathscr{A}$, there exists a negligible function μ, such that the probability of distinguishing two permutations in a CPA indistinguishability experiment is less than $1/2+\mu$.

3.4 Secure Transformation and Permutation Schemes for Matrix and Vector Privacy

In order to securely outsource a CSP problem to the cloud, the client must first conceal the private data by performing certain computations. To this end, we describe secure transformation, permutation, and vector addition algorithms that conceal the non-zero elements and structure of a private matrix and a vector, which can preserve the privacy of the CSP problem.

3.4.1 Secure Matrix Multiplications

We propose that the client can efficiently conceal the non-zero values of a private matrix by employing sparse random matrix multiplications. Specifically, consider a private matrix $\mathbf{Q} \in \mathbb{R}^{m\times n}$, with non-zero elements $q_{i,j} \leftarrow \{0,1\}^K$ for $(i,j) \in \mathscr{S}_{\mathbf{Q}}$, where $\mathscr{S}_{\mathbf{Q}}$ is the structure of $\mathbf{Q}$ as defined in (3.3). We assume that the elements of $\mathbf{Q}$ are within the range $[-G, G]$, where $G = 2^s$ $(s > 0)$ is a positive constant, and that matrix $\mathbf{Q}$ has non-zero diagonal elements and at least one non-zero off-diagonal element per column. The client can hide $\mathbf{Q}$'s non-zero values by performing the following matrix multiplication:

$$\tilde{\mathbf{Q}} = (\mathbf{I} + \mathbf{FD})\mathbf{Q}, \tag{3.5}$$

where $\mathbf{I} \in \mathbb{R}^{m\times m}$ is the identity matrix. Matrix $\mathbf{F} \in \mathbb{R}^{m\times m}$ is a diagonal matrix, i.e.,

$$f_{i,j} = \begin{cases} t_i, & i = j \text{ for } i,j \in [1,m] \\ 0, & \text{otherwise} \end{cases},$$

where t_i is the ith element of vector $\mathbf{t} \in \mathbb{R}^{n\times 1}$ and determined by a pseudorandom function $F_c : \{0,1\}^w \times \{0,1\}^w \to \{0,1\}^w$, i.e.,

$$t_i = F_c(r_i, g) \; \forall i \in [1, m], \tag{3.6}$$

where r_i is a random string and g is a constant one. The elements of vector $\mathbf{t}$ are within the range $(0,1]$. Matrix $\mathbf{D} \in \mathbb{R}^{m\times m}$ is defined by

$$d_{i,j} = \begin{cases} v_{i,j}, & \text{if } (i,j) \in \mathscr{K} \\ 0, & \text{otherwise} \end{cases}$$

for $i, j \in [1, m]$, where $v_{i,j}$'s are arbitrary constants with absolute value ranging from $v_{min} = 2^{s+y}$ to $v_{max} = 2^{s+y+z}$ $(y \le 1, z \le 1)$. The set of index pairs $\mathscr{K}$ is given by

$$\mathscr{K} = \{\exists (i,j) \mid d_{i,j} q_{j,i} \neq 0 \wedge \; i \neq j \text{ for } i \in [1,m]\}.$$

Consequently, the non-zero elements of $\tilde{\mathbf{Q}}$ in (3.5) are given by

$$\tilde{q}_{i,j} = q_{i,j} + t_i \sum_{(i,k)\in\mathscr{K}} v_{i,k} q_{k,j} \tag{3.7}$$

for $i \in [1, m]$ and $j \in [1, n]$. Note that $\tilde{q}_{i,j}$ is within the range $[-G-L, G+L]$, where $L = \sum_{(i,k)\in\mathscr{K}} v_{i,k} q_{k,j}$.

We denote these computations as

$$Mask^{F_c}(r_i, q_{i,j}) = \tilde{q}_{i,j}. \tag{3.8}$$

We can now arrive at a theorem about the CPA-security in value of the matrix multiplications in (3.5).

Theorem 2. If $F_c(\cdot,\cdot)$ is a pseudorandom function, the matrix multiplications in (3.5) are computationally indistinguishable in value under a CPA.

Proof. According to Definition 3, we need to show that given two arbitrary matrices $\mathbf{Q}^1$ and $\mathbf{Q}^2$ with the same structure as $\mathbf{Q}$, i.e., $\mathscr{S}_{\mathbf{Q}^1} = \mathscr{S}_{\mathbf{Q}^2} = \mathscr{S}_{\mathbf{Q}}$, 1) $\tilde{\mathbf{Q}}^1$ and $\tilde{\mathbf{Q}}^2$ have the same structure; and 2) that $\tilde{q}^1_{i,j}$ and $\tilde{q}^2_{i,j}$ $(\forall(i,j) \in \mathscr{S}_{\tilde{\mathbf{Q}}})$ are indistinguishable under a CPA.

To prove 1), we show that both $\tilde{\mathbf{Q}}^0$ and $\tilde{\mathbf{Q}}^1$ have the same structure. Since $\mathscr{S}_{\mathbf{D}_0} = \mathscr{S}_{\mathbf{D}_1}$ and $\mathscr{S}_{\mathbf{Q}_0} = \mathscr{S}_{\mathbf{Q}_1}$, we can see that $\mathscr{S}_{\tilde{\mathbf{Q}}_0} = \mathscr{S}_{\tilde{\mathbf{Q}}_1}$, i.e., both $\tilde{\mathbf{Q}}_0$ and $\tilde{\mathbf{Q}}_1$ have the same structure.

To prove 2), we need to show that a probabilistic polynomial time distinguisher D cannot distinguish $\tilde{q}^0_{i,j}$ from $\tilde{q}^1_{i,j}$ for any $(i,j) \in \mathscr{S}_{\tilde{\mathbf{Q}}_t}$ (for $s \in [1,2]$) with a probability significantly higher than 1/2 in a CPA experiment.

Specifically, suppose a probabilistic polynomial-time adversary $\mathscr{A}$ carries out a CPA indistinguishability experiment $\mathscr{A}^{F_c}$ as shown in Algorithm 2. In particular, an adversary $\mathscr{A}$ outputs two arbitrary numbers, $q^0_{i,j}, q^1_{i,j} \leftarrow \{0,1\}^K$. A bit $b \leftarrow \{0,1\}$ is randomly chosen, and $Mask^F(r'_i, t_j, q^b_{i,j}) = \tilde{q}^b_{i,j}$ is computed and given to $\mathscr{A}$, where r'_i is a random number. $\mathscr{A}$ has oracle access to $Mask^{F_c}$ and eventually outputs b'. If $b' = b$, we say that $\mathscr{A}$ succeeds and set $\mathscr{A}^{F_c} = 1$. Note that the random number r'_i is known to the adversary.

Algorithm 2 A CPA Indistinguishability Experiment: $\mathscr{A}^{F_c}$

```
An adversary $\mathscr{A}$ outputs two arbitrary numbers $q^0_{i,j}, q^1_{i,j} \leftarrow \{0,1\}^K$.
A random bit $b \leftarrow \{0,1\}$ is chosen. $Mask^{F_c}(r'_i, t_j, q^b_{i,j}) = \tilde{q}^b_{i,j}$ is computed and given to
    $\mathscr{A}$, where $r'_i$ is randomly chosen.
$\mathscr{A}$ continues to have oracle access to $Mask^{F_c}$ and outputs a bit $b'$.
Ensure: 1 if $b' = b$ and 0 otherwise.
```

Now consider an experiment $\mathscr{A}^{f_c}$ that is exactly the same as $\mathscr{A}^{F_c}$ except that a truly random function $f_c : \{0,1\}^w \rightarrow \{0,1\}^w$ is used in place of F_c. $\mathscr{A}$'s probability of success, i.e., $\mathscr{A}^{f_c} = 1$, depends on two cases:

(1) The oracle chooses the same random number r'_i used to compute $\tilde{q}^b_{i,j}$ to answer at least one of $\mathscr{A}$'s queries. In this case, $\mathscr{A}$ can easily tell which of its values was masked and hence correctly get $b' = b$, i.e., $A^{f_c} = 1$. We denote this case as C_0.

(2) The oracle never chooses the same random number r'_i used to compute $\tilde{q}^b_{i,j}$ to answer $\mathscr{A}$' queries. In this case, the adversary $\mathscr{A}$ succeeds with a negligible probability. We denote this case as C_1.

In particular, recall that $\tilde{q}^b \in [-L-G, G+L]$, and hence $\tilde{q}^b_{i,j} \in [-e2^{\kappa}, e2^{\kappa}]$, where $\kappa = s+y+z+1$. The best strategy for an adversary $\mathscr{A}$ is to set $q^0_{i,j} = 0$ and $|q^1_{i,j}| = G$, and return $b \leftarrow \{0,1\}$ with equal probability if $-L_i \le \tilde{q}^b_{i,j} \le L_i$, and 1 if $\tilde{q}^b_{i,j} < -L_i$ or $\tilde{q}^b_{i,j} > L_i$. Therefore, we have that the success probability of the distinguisher is given by

$$
\begin{aligned}
& Pr[\mathscr{A}^{f_c} = 1|C_1] \\
& = \frac{1}{2} Pr[-L_i \le \tilde{q}^b_{i,j} \le L_i] \\
& + Pr[\tilde{q}^b_{i,j} < -L_i] + Pr[\tilde{q}^b_{i,j} > L_i] \\
& = \frac{1}{2}\Big(1 - Pr[\tilde{q}^b_{i,j} < -L_i] - Pr[\tilde{q}^b_{i,j} > L_i]\Big) \\
& + Pr[\tilde{q}^b_{i,j} < -L_i] + Pr[\tilde{q}^b_{i,j} > L_i]
\end{aligned}
$$

where

$$
\begin{aligned}
&Pr[\tilde{q}^b_{i,j} > L_i] \\
&= Pr[q^b_{i,j} + t_i \sum_{(i,k)\in\mathscr{K}} v_{i,k} q_{k,j}) > L_i] \\
&= Pr[t_i > \frac{L_i - \tilde{q}^b_{i,j}}{\sum_{(i,k)\in\mathscr{K}} v_{i,k} q_{k,j}}] \\
&\leq Pr[t_i > \frac{L_i - G}{L_i}] \\
&= Pr[t_i > 1 - \frac{G}{L_i}] \\
&= \frac{G}{L_i}
\end{aligned}
$$

Similarly, we find that $Pr[\tilde{q}^b_{i,j} < -2^u] \leq \frac{G}{L_i}$. Consequently, we have that the probability of success for adversary $\mathscr{A}$ in case C_1 is bounded as follows:

$$0 < Pr[\mathscr{A}^{f_c} = 1|C_1] \leq \frac{1}{2} + \frac{G}{L_i}.$$

Note that $K = 2^u$ and $L_i \in [e2^{u+s}, e2^{u+s+z}]$. Thus, we have

$$\nu(\kappa) = \frac{G}{L_i} \leq \frac{2^u}{e2^{u+s}} = \frac{1}{e2^{\kappa-u-s-1}},$$

which is a negligible function.

Therefore, the probability of $\mathscr{A}^{f_c} = 1$, i.e., $\mathscr{A}$ succeeding, can be calculated as:

$$
\begin{aligned}
&Pr[\mathscr{A}^{f_c} = 1] \\
&= Pr[\mathscr{A}^{f_c} = 1|C_0]Pr[C_0] + Pr[\mathscr{A}^{f_c} = 1|C_1]Pr[C_1] \\
&\leq Pr[C_0] + Pr[\mathscr{A}^{f_c} = 1|C_1]
\end{aligned}
$$

Since $\mathscr{A}$ is a polynomial time adversary, it can at most make $\alpha(w)$ queries to the oracle, where $\alpha(\cdot)$ is a polynomial function. Hence, in $\mathscr{A}^{f_c}$, $\mathscr{A}$ can query the oracle at most $\alpha(w)$ times. Considering that the values returned by the oracle to $\mathscr{A}$ are truly random numbers, the probability that $\mathscr{A}$ succeeds, i.e., $\mathscr{A}^{f_c} = 1$, is

$$Pr[\mathscr{A}^{f_c} = 1] \leq \frac{\alpha(w)}{2^w} + \nu(w) + \frac{1}{2} \tag{3.9}$$

Next, we define the function μ as follows:

$$\mu(w) = Pr[\mathscr{A}^{F_c} = 1] - (\nu(w) + \frac{1}{2}),$$

and hence we have

$$Pr[\mathscr{A}^{F_c} = 1] = \frac{1}{2} + \nu(w) + \mu(w). \tag{3.10}$$

Intuitively, if $\mu(w)$ is not negligible, then the difference between (3.9) and (3.10) is also not negligible. Thus, an adversary $\mathscr{A}$ would be able to distinguish a truly random function and a pseudorandom function.

To formally prove this, we use $\mathscr{A}$ to construct a distinguisher D. To this end, D emulates the CPA indistinguishability experiment for $\mathscr{A}$ as described in Algorithm 3 and observes whether $\mathscr{A}$ succeeds or not. If $\mathscr{A}$ succeeds, D guesses that its input is a pseudorandom function, while if $\mathscr{A}$ fails, D guesses that this oracle is a truly random function.

We observe that if D's oracle uses a truly random function, the view of $\mathscr{A}$ when called by D as a sub-routine is identical to its view when called by $\mathscr{A}^{f_c}$. Therefore, we have that

$$Pr[D^{f_c} = 1] = Pr[\mathscr{A}^{f_c} = 1]. \tag{3.11}$$

Algorithm 3 Distinguisher D

```
D is given access to an oracle $\mathbb{O}$.
When $\mathscr{A}$ queries with two arbitrary numbers $q^0_{i,j}, q^1_{i,j}$, choose a random bit $b \leftarrow \{0,1\}$, compute $\tilde{q}^b_{i,j} = v_i t_j q^b_{i,j}$ where $v_i$ is the output of the oracle $\mathbb{O}$ and $t_j$ is as defined before, and return it to $\mathscr{A}$.
Continue answering any oracle queries of $\mathscr{A}$. Eventually, $\mathscr{A}$ outputs $b'$.
Ensure: 1 if $b' = b$ and 0 otherwise.
```

On the other hand, if D's oracle is a pseudorandom function, then the view of $\mathscr{A}$ when called by D is identically distributed to its view when called by $\mathscr{A}^{F_c}$. Thus, we get

$$Pr[D^{F_c} = 1] = Pr[\mathscr{A}^{F_c} = 1]. \tag{3.12}$$

Taking the difference of equations (3.12) and (3.11), we get

$$Pr[D^{F_c} = 1] - Pr[D^{f_c} = 1] \geq \mu(w) - \frac{\alpha(w)}{2^w}$$

Since we have assumed that F_c is a pseudorandom function, the term $\mu(w) - \frac{\alpha(w)}{2^w}$ must be negligible by Definition 3. Moreover, since $\alpha(w)$ is polynomial, this implies that $\mu(w)$ must also be negligible, making our value masking transformation secure under CPA.

By union bound, this concludes the proof. □

3.4.2 Secure Matrix Permutations

Although the matrix transformation in equation (3.5) hides the values of the non-zero elements in **Q**, it still reveals their original positions, i.e., **Q**'s structure, which is also private. Next, we design secure permutations that can hide **Q**'s structure by randomly reordering the rows and columns of $\tilde{\mathbf{Q}}$.

To randomly permute $\tilde{\mathbf{Q}}$'s row index vector $\mathbf{e} \in \mathbb{R}^{m\times 1}$, the client computes the following:

$$\mathbf{e}' = \mathscr{M}(\mathbf{e}), \;\; \hat{\mathbf{e}}' = F(\mathbf{r}, \mathbf{e}'), \;\; \hat{\mathbf{e}} = \mathscr{M}^{-1}(\hat{\mathbf{e}}') \tag{3.13}$$

where $\mathscr{M} : \mathbb{R}^m \rightarrow \{0,1\}^k$ ($k = \lceil \log_2 m! \rceil$) is a function that maps index vectors to bit strings, $F : \{0,1\}^k \rightarrow \{0,1\}^k$ is a pseudorandom permutation, $\mathbf{r} \in \{0,1\}^k$ is a random bit string, and $\mathscr{M}^{-1} : \{0,1\}^k \rightarrow \mathbb{R}^m$ is the inverse of $\mathscr{M}$. We denote these computations as

$$Perm^F(\mathbf{r}, \mathbf{e}) = \hat{\mathbf{e}}. \tag{3.14}$$

Similarly, we can denote by $Perm^F(\mathbf{r}', \mathbf{u})$ the random permutation of column index vector $\mathbf{u} \in \mathbb{R}^n$, where $\mathbf{r}' \in \{0,1\}^{k'}$ ($k' = \lceil \log_2 n! \rceil$) is a random bit string.

The client applies the random permutations $Perm(\mathbf{r}, \mathbf{e})$ and $Perm(\mathbf{r}', \mathbf{u})$ to $\tilde{\mathbf{Q}}$ through the following multiplications:

$$\hat{\mathbf{Q}} = \mathbf{E}\tilde{\mathbf{Q}}\mathbf{U} \tag{3.15}$$

where $\mathbf{E} \in \mathbb{R}^{m\times m}$ and $\mathbf{U} \in \mathbf{R}^{n\times n}$ are random permutation matrices, and their elements are defined by

$$\begin{aligned} e_{i,j} &= \delta_{\pi(i),j} \quad \forall i \in [1,m], j \in [1,m] \\ u_{i,j} &= \delta_{\pi(i),j} \quad \forall i \in [1,n], j \in [1,n] \end{aligned}$$

where i and j are the row and column indexes, respectively, and the function $\pi(\cdot)$ maps an original index i to its permuted index, i.e., $\pi(i) = \hat{e}_i$ (for $i \in [1,m]$) and $\pi(i) = \hat{u}_i$ (for

$i \in [1, n]$). Besides, $\delta_{i,j}$ is the Kronecker delta function given by

$$\delta_{i,j} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} .$$

The cloud is able to recover the original matrix by applying the inverse permutations, i.e.,

$$\tilde{\mathbf{Q}} = \mathbf{E}^\top \hat{\mathbf{Q}} \mathbf{U}^\top \tag{3.16}$$

where $\top$ denotes the matrix transpose operation. To reach this result, we have used the orthogonal property of permutation matrices, i.e, $\mathbf{E}^\top \mathbf{E} = \mathbf{I}$ and $\mathbf{U}\mathbf{U}^\top = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix.

We now state a theorem about the CPA-security in structure of the above matrix permutations in (3.15).

Theorem 3. If $F_c(\cdot, \cdot)$ is a pseudorandom function, then the row and column permutations described in (3.15) are computationally indistinguishable in structure under a CPA.

Proof. The proof follows a similar approach to that in the proof of Theorem 2. The main difference is that, instead of using the pseudorandom function to generate the values of the concealing matrix, we use it to find a permutation of the rows (or columns) of the private matrix. Since the permutations are randomly chosen, the CPA security holds. Due to space limit, we omit the proof. □

3.4.3 Secure Vector Additions

Besides protecting the matrices, we also need to protect the vectors in **P1**, i.e., x_j's. To this end, we propose a secure vector addition scheme. In particular, the client hides the private variable vector $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ by adding a randomly generated vector, denoted by $\mathbf{r} \in \mathbb{R}^{n\times 1}$ as follows:

$$\mathbf{y} = \mathbf{x} + \mathbf{r}, \tag{3.17}$$

where $y_j = x_j + r_j$ for any $j \in [1, n]$, and y_j, x_j and r_j are the jth element of vector **y**, **x**, and **r**, respectively. Here we assume that x_j is in the range $[-K, K]$ where $K = 2^k (k > 0)$ is a positive constant. In addition, r_j $(j \in [1, n])$ is uniformly distributed on $[-L, L]$ with the corresponding probability density function as follows:

$$f_r(r_j) = \begin{cases} \frac{1}{2L}, -L \leq r_j \leq L \\ 0, \text{otherwise} \end{cases} \tag{3.18}$$

where $L = 2^{k+l} (l > 0)$ is a positive constant. We obtain the following theorem that vectors **r** and **y** are computationally indistinguishable.

Theorem 4. If $F_c(\cdot, \cdot)$ is a pseudorandom function, the vector additions in (3.17) are computationally indistinguishable in value under a CPA.

PROOF. According to Definition 2, we need to prove that any polynomial-time distinguisher D cannot distinguish y_j from r_j for $j \in [1, n]$ except with non-negligible success probability, where y_j and r_j are the jth element of the vector **y** and **r** respectively. The best strategy for a polynomial-time distinguisher D is to follow the rules: D outputs 0 or 1 with the same probability of $\frac{1}{2}$ if the chosen element, i.e., z_j, is within the range

$[0, L]$. Moreover, the distinguisher D will only output 1 if z_j is in the range $(-\infty, 0)$ or (L, ∞).

Suppose that the element $y_j = x_j + r_j$ is chosen from the vector $\mathbf{y}$, the success probability of the distinguisher is obtained by:

$$\begin{aligned} Pr \quad & [D(y_j) = 1] \\ = \quad & \frac{1}{2} Pr[0 \le x_j + r_j \le L] \\ + \quad & Pr[x_j + r_j < 0] + Pr[x_j + r_j > L] \\ = \quad & \frac{1}{2}[1 - Pr[x_j + r_j < 0] - Pr[x_j + r_j > L]] \\ + \quad & Pr[x_j + r_j < 0] + Pr[x_j + r_j > L] \end{aligned} \tag{3.19}$$

Recall that x_j is in the range $[-K, K]$ and that r_j is sampled from a uniform distribution with probability density function in (3.18). We have that

$$\begin{aligned} & Pr[x_j + r_j > L] \\ & = Pr[r_j > L - x_j] \le Pr[r_j > L - K] = \frac{K}{L}. \end{aligned} \tag{3.20}$$

Similarly, we can have that

$$Pr[x_j + r_j < 0] = Pr[r_j < -x_j] \le Pr[r_j < K] = \frac{K}{L}. \tag{3.21}$$

Consequently, the success probability of the distinguisher D follows the inequality:

$$Pr[D(y_j) = 1] \le \frac{1}{2} + \frac{K}{L}. \tag{3.22}$$

On the other hand, suppose the element r_j is chosen from the vector **r**, by following the procedures above and we also obtain that:

$$Pr[D(r_j) = 1] = \frac{1}{2}. \tag{3.23}$$

According to equation 3.4 in Definition 2, for $\forall j \in [1, n]$, we know that

$$|Pr[D(y_j) = 1] - Pr[D(r_j) = 1]| \leq \frac{K}{L}. \tag{3.24}$$

Note that $K = 2^k$ and $L = 2^{k+l}$. Thus, we can obtain that

$$\mu(l) = \frac{K}{L} \leq \frac{2^k}{2^{k+l}} = \frac{1}{2^l} \tag{3.25}$$

Since we can assign a large value to l, function $\mu(l)$ become negligible. By union bound, we complete the proof for Theorem 4. □

3.5 Secure Outsourcing Scheme Design

In this section, we develop a secure outsourcing scheme for large-scale CSPs. Note that the original CSP problem **P1** is a nonlinear problem. Our main idea is to firstly linearize the nonlinear functions in **P1** with arbitrary accuracy and obtain a series of linear programming problems denoted by **P2**. After that, we propose the secure outsourcing scheme for solving the large-scale CSPs.

3.5.1 Linearization of a General Nonlinear Function

Consider a continuous nonlinear convex function $h(t)$ where $t \in [t_a, t_b]$. We use a linear approximation function, i.e., $\hat{h}(\cdot)$, to approximate the original function $h(t)$. Specifically, by inserting k grid points, denoted by $\{t_v | v = 1, \cdots, k\}$, a continuous nonlinear function $h(t)$ can be approximated by[47]:

$$\text{for} \quad t = \sum_{v=1}^{k} t_v \lambda_v, \quad h(t) \approx \sum_{v=1}^{k} h(t_v)\lambda_v, \tag{3.26}$$

where $\lambda_v (v \in [1, k])$ is the coefficient for the grid point t_v, and

$$\sum_{v=1}^{k} \lambda_v = 1, \lambda_v \geq 0, v = 1, \cdots, k. \tag{3.27}$$

3.5.2 Linearization of Problem P1

Based on the linearization method presented in (3.26) and (3.27), we can transform the original problem **P1** into the problem **P2**, i.e.,

$$\begin{aligned}
\textbf{P2:} \quad & \min_{\{\lambda_{jv} | j \in [1,n], v \in [1,k_j]\}} \quad \sum_{j=1}^{n} \sum_{v=1}^{k_j} f_j(\hat{x}_{jv})\lambda_{jv}, \\
\textbf{s.t.} \quad & \sum_{j=1}^{n} \sum_{v=1}^{k_j} g_{ij}(\hat{x}_{jv})\lambda_{jv} \leq b_i, i = 1, \cdots, m \qquad (3.28) \\
& \sum_{v=1}^{k_j} \lambda_{jv} = 1, j = 1, \cdots, n \qquad (3.29) \\
& \lambda_{jv} \geq 0, v = 1, \cdots, k_j, j = 1, \cdots, n \qquad (3.30)
\end{aligned}$$

where k_j is the number of grid points for the variable x_j, and x_{jv}'s $(v \in [1, k_j])$ are the grid points for the variable x_j. Since **P2** is a linear programming problem, we can solve it with existing techniques such as interior point methods[84].

3.5.3 An Optimal Solver for the Original Large-scale CSP

Since the accuracy of the linear approximations for separable problem heavily depends on the number of the grid points for each variable, there is a tradeoff between the accuracy and the convergence speed. That is, when we increase the number of the grid points to improve the approximation accuracy, the size of the approximation problem **P2** increases dramatically, hence increasing the complexity of the approximation problem. Considering that the problem is already a large-scale problem, how to optimally choose the number of grid points is very critical. Previous works only simply add random number of grid points and divide the range of x's into same-size subrange for each grid point, which only results in suboptimal accuracy for the linear approximation. In what follows, we describe how to find the optimal number of the grid points so that we can achieve arbitrary accuracy of the linear approximation.

We solve this problem in an iterative manner. Assume that at dth iteration, we solve the **P2** and let $\hat{\lambda}_{jv}$'s be the optimal solution to **P2**. Furthermore, let $s_i \geq 0$ and t_j be the optimal Lagrangian multipliers for constraints (3.28) and (3.29), respectively. Then the solution set can be denoted by $\Omega = \{\hat{\lambda}_{jv}, s_i, t_j | i \in [1, m], j \in [1, n], v \in [1, k_j]\}$. Next, the question is whether adding a new grid point can achieve a better linear approximation and the minimum objective function value would further decrease. Therefore, we have the following theorem.

Theorem 5. Let $\Omega = \{\hat{\lambda}_{jv}, s_i, t_j | i \in [1, m], j \in [1, n], v \in [1, k_j]\}$ be the solution set to the problem **P2** and $\hat{x}_{jv}$'s, $(v \in [1, k, j = 1, \cdots, n)$ be the corresponding grid points. Consider that functions f_j and g_{ij} are convex functions. Denote by $\psi_j(\hat{x}_j)$ a function as follows:

$$\psi_j(\hat{x}_j) = f_j(\hat{x}_j) + \sum_{i=1}^{m} s_i g_{ij}(\hat{x}_j) + t_j \tag{3.31}$$

for $j = 1, \cdots, n$, where $\hat{x}_j = \sum_{v=1}^{k} \hat{x}_{jv}\hat{\lambda}_{jv}$. Then we have that

(1) If $\forall j = 1, \cdots, n, \psi_j(\hat{x}_j) \geq 0$, then $\Omega = \{\hat{\lambda}_{jv}, s_i, t_j\}, i = 1, \cdots, m, j = 1, \cdots, n$ is an optimal solution to problem **P1**, and the optimal objective function value is $\sum_{j=1}^{n} f_j(\hat{x}_j)$.

(2) Otherwise, if $\psi_j(\hat{x}_j) \leq 0$, denote $\hat{x}_{v_0 j}$ to be the corresponding optimal solution to (3.31) and set it as the new grid point. Then we will obtain a new approximating linear programming problem with a minimum objective value not higher than $\sum_{j=1}^{n} f_j(\hat{x}_j)$.

Proof. Due to the space limits, we omit the proof and please refer to[82] for detailed proof. □

Theorem 5 helps us determine the optimal number of grid points to find the final solution. The whole algorithm for solving the CSP problem is summarized as Algorithm 4.

Algorithm 4 An Efficient Solver for CSP Problem

```
Require: P2, initial grid points $\hat{x}_{j0}$ and $k_j = 1$
Solve P2
Solve subproblem (3.31) and obtain $\psi_j(\hat{x}_j)$'s and $\hat{\lambda}_{jv}$'s ($v \in [1, k_j]$)
For ($\psi_j(\hat{x}_j) < 0$)
    Add new grid point $\hat{x}_{j(k_j+1)} = \sum_{v=1}^{k_j} \hat{x}_{jv}\hat{\lambda}_{jv}$ with $\hat{\lambda}_{j(k_j+1)} = 0$, and set $k_j = k_j + 1$
    Update P2, solve P2 and obtain $\hat{\lambda}_{jv}, v = 1, \cdots, k_j, j = 1, \cdots, n$
    Solve subproblem (3.31) and update $\psi_j(\hat{x}_j)$ for all $j \in [1, n]$
    end
Ensure: $\hat{x}_j = \sum_{v=1}^{k_j} \hat{x}_{jv}\hat{\lambda}_{jv}$ and $\sum_{j=1}^{n} f_j(\hat{x}_j)$
```

3.5.4 An Efficient Secure Outsourcing Algorithm

In what follows, we develop an efficient secure outsourcing algorithm to solve the large-scale CSP problem with the help of the cloud.

Particularly, as shown in Section 3.5.2, the client linearizes the original problem **P1** into **P2**. Considering that **P2** is a large-scale problem and computationally prohibitive for the client to solve by itself, **P2** will be outsourced to the cloud for solutions. To protect client's data privacy, we conduct some transformations based on the proposed schemes in Section 3.4.

Before we delve into the secure outsourcing algorithm, we rewrite the **P2** in a vector form denoted as **P2***:

$$\textbf{P2*} \qquad \textbf{Min} \quad \mathbf{f}^T\lambda,$$

$$\textbf{s.t.} \qquad \mathbf{G}\lambda \le \mathbf{b} \tag{3.32}$$

$$\mathbf{H}\lambda = \mathbf{1} \tag{3.33}$$

$$\lambda \ge \mathbf{0} \tag{3.34}$$

where $\mathbf{f} = \{f_1(x_{1_1}), \cdots, f_1(x_{1k_1}), \cdots, f_j(x_{j1}), \cdots, f_j(x_{jk_j})\}$ and $\lambda = \{\lambda_{11}, \cdots, \lambda_{1k_1}, \cdots, \lambda_{j1}, \cdots, \lambda_{jk_j}\}$ are vectors. $\mathbf{G}$ is a $m \times p$ matrix where p is the total number of grid points, i.e., $= \sum_{i=1}^{j} k_i$. $\mathbf{b} = \{b_i\}, i \in [1, m]$, and $\mathbf{H}$ is a $j \times p$ matrix whose elements, denoted by $h_{uv}, i = 1, \cdots, j = 1, \cdots, j$, is:

$$h_{uv} = \begin{cases} 1, \sum_{i=1}^{u-1} k_u < v \le \sum_{i=1}^{u} k_u \\ 0, \text{otherwise.} \end{cases} \tag{3.35}$$

$\mathbf{1}$ is a $n \times 1$ vector whose elements are 1 and $\mathbf{0}$ is a $p \times 1$ vector whose elements are 0. As discussed in Section 3.3.2, in order to protect the data privacy, we not only need to conceal the coefficients, i.e., $\mathbf{f}$, $\mathbf{G}$ and $\mathbf{H}$, but also the output vector λ. Therefore, based on the proposed schemes in Section 3.4, we can transform the problem **P2*** into a secure problem **P3** in the following:

$$\textbf{P3:} \quad \textbf{Min} \quad \hat{\mathbf{f}}^T \hat{\lambda},$$

$$\textbf{s.t.} \quad \hat{\mathbf{G}}\hat{\lambda} \le \hat{\mathbf{b}} \tag{3.36}$$

$$\hat{\mathbf{H}}\hat{\lambda} = \hat{\mathbf{1}} \tag{3.37}$$

$$\hat{\mathbf{I}}\hat{\lambda} \ge \hat{\mathbf{0}} \tag{3.38}$$

where $\hat{\lambda} = \mathbf{N}^{-1}(\lambda + \mathbf{r})$ and $\mathbf{r}$ is a $p \times 1$ random vector, $\hat{\mathbf{f}} = \gamma \mathbf{N}^T \mathbf{c}$ and γ is a random number. $\hat{\mathbf{G}} = \mathbf{MGN}$ and $\hat{\mathbf{H}} = \mathbf{M}'\mathbf{G}\mathbf{N}'$ where $\mathbf{M}$, $\mathbf{N}$,$\mathbf{M}'$, and $\mathbf{N}'$ are random dense matrices. $\hat{\mathbf{b}} = \mathbf{M}(\mathbf{b} + \mathbf{Gr})$ (with $\mathbf{b} + \mathbf{Gr} \neq \mathbf{0}$). Regarding the lower bound constraint (3.34), we propose to set $\hat{\mathbf{I}} = (\mathbf{I} - \tau \mathbf{MG})\mathbf{N}$ where τ is a matrix such that $\tau \hat{\mathbf{b}} = \mathbf{Ir}$ and $\mathbf{I}$ is the identity matrix.

After transforming the **P2*** into **P3**, the client can outsource this secure problem to the cloud. The cloud solves **P3** and its Lagrange dual problem, then sends the results back to the client. The client can obtain the final results by computing $\lambda^* = \mathbf{N}\hat{\lambda}^* - \mathbf{r}$. To prevent the malicious cloud from cheating the results, the client can verify the correctness by checking whether the objective value of **P3** equals the Lagrange dual problem[85]. The details of the proposed secure outsourcing scheme for large-scale CSP problem can be summarized in Algorithm 5.

Algorithm 5 A Secure Outsourcing Scheme for CSP Problem

```
Require: P3, initial grid point $\hat{x}_{j0}$ and $k_j = 1$
Cloud solves P3 and sends the result to the client
Client solves $\Lambda_{jk_j} = \bar{\Lambda}_{jk_j} - \mathbf{r_j}, j = 1, \cdots, n$
Client solves subproblem (3.31) and obtain $\psi_j(\hat{x}_j)$'s and $\hat{\lambda}_{jv}$'s $(v \in [1, k_j])$
For $(\psi_j(\hat{x}_j) < 0)$
    Client adds a new grid point $\hat{x}_{j(k_j+1)} = \sum_{v=1}^{k_j} \hat{x}_{jv}\hat{\lambda}_{jv}$ with $\hat{\lambda}_{j(k_j+1)} = 0$, and set $k_j = k_j + 1$
    Client updates P2* with the new grid points
    Client transforms P2* into P3 and send it to the cloud
    Cloud solves P3 and sends the result to the client
    Client solves subproblem (3.31) and update $\psi_j(\hat{x}_j)$ for all $j \in [1, n]$
    end
Ensure: $\hat{x}_j = \sum_{v=1}^{k} \hat{x}_{jv}\hat{\lambda}_{jv}$ and $\sum_{j=1}^{n} f_j(\hat{x}_j)$
```

3.6 Theoretical Analysis of the Proposed Algorithm

In this section, from a theoretical perspective, we provide both the correctness analysis and privacy analysis of our secure outsourcing scheme.

3.6.1 Correctness Analysis

We can arrive at a theorem about the correctness of our secure outsourcing scheme.

Theorem 6. The proposed secure outsourcing scheme in Algorithm 5 gives the optimal solution to the original CSP problem **P1**.

Proof. Our proposed secure outsourcing scheme is an interactive algorithm that requires the client and the cloud to cooperate and solve the original CSPs. In particular, by adding a number of grid points, the original problem **P1** is linearized into **P2** by the client. In order to protect the data privacy, the client transforms **P2** into **P3** by the secure schemes in Section 3.4. We can easily transform **P3** back to **P2** by substracting the added vectors and matrices. The solutions to **P3** are obtained by the cloud and transferred back to the client. The client can obtain the final results by computing $\lambda^* = \mathbf{N}\hat{\lambda}^* - \mathbf{r}$. As λ^*'s are the solutions to **P2**, they are the solutions to **P1** as well, which concludes the proof. □

3.6.2 Privacy Analysis

Inspecting the proposed secure outsourcing algorithm, we observe that the cloud only has access to the securely transformed linear programs, and hence it is unable to learn private information from the client.

Specifically, in the process of securely outsourcing the dth iteration's linear program as in Section 3.5.4, the client shares the cloud with the transformed matrices $\hat{\mathbf{G}}$, $\hat{\mathbf{H}}$, $\hat{\mathbf{I}}$, and the transformed vectors $\hat{\mathbf{f}}$, $\hat{\mathbf{b}}$, and $\hat{\mathbf{1}}$. According to Theorem 2, Theorem 3 and Theorem 4, the transformed matrices and vectors are computationally indistinguishable both in value and in structure under a CPA. Thus, the cloud cannot derive any information about the elements of the original linear program's matrices **G**, **H**, **I**, **f**, **b**, **1**, from the transformed matrices that the client uploads. Similarly, in the process of solving the transformed linear program at dth iteration, the cloud obtains the concealed solution vector $\hat{\lambda}$, which according to Theorem 4, is CPA secure as well.

Moreover, since the client locally linearizes the original CSP, the cloud is unable to determine the objective and coefficient functions, i.e., $f_j, g_{ij}, i = 1, \cdots, m, j = 1, \cdots, n$.

3.7 Evaluation Results

In this section, we present the performance of the proposed scheme for secure outsourcing of large-scale CSPs.

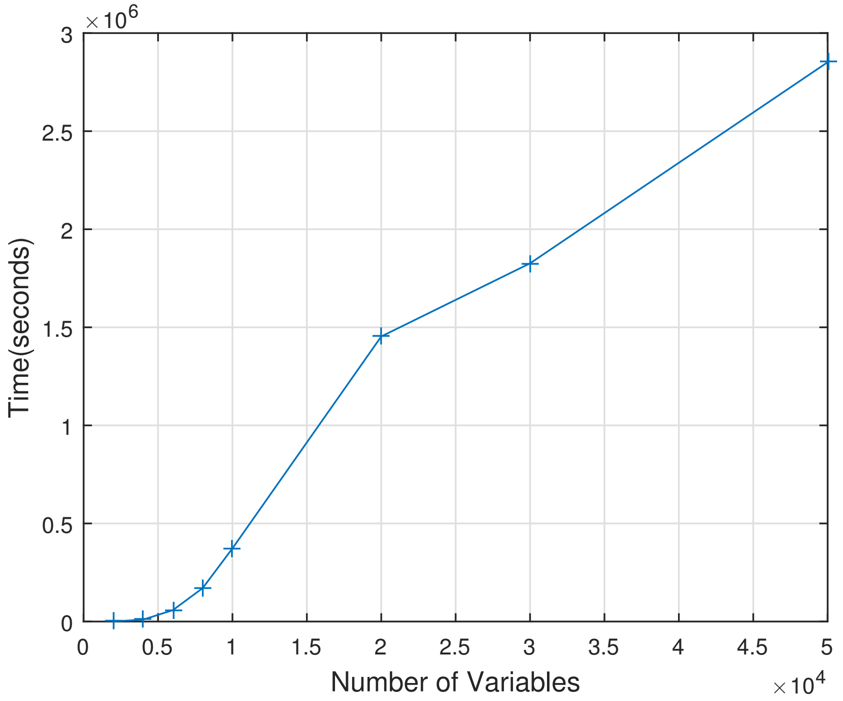

Figure 3.2. Computing time at the client of the proposed algorithm at the client and cloud for different separable program sizes.

3.7.1 Experiment Setup

To evaluate our proposed algorithm in a practical scenario, we implement the client-side computations of the proposed algorithm on a laptop with a dual-core 2.6GHz CPU, 8GB RAM, and a 150GB solid state drive, and the cloud-side computations on an Amazon Web Services (AWS) Elastic Computing Cloud (EC2) instance. Amazon Elastic Computing Cloud provides scalable computing capacity for large-scale computations[86]. We implement both the client-side and the cloud computations of the proposed algorithm on Matlab 2015a. We evaluate the performance of our algorithm by generating random large-scale CSPs with the number of variables ranging from 2×10^3 to 5×10^4. We set the objective functions and constraints equal to quadratic functions.

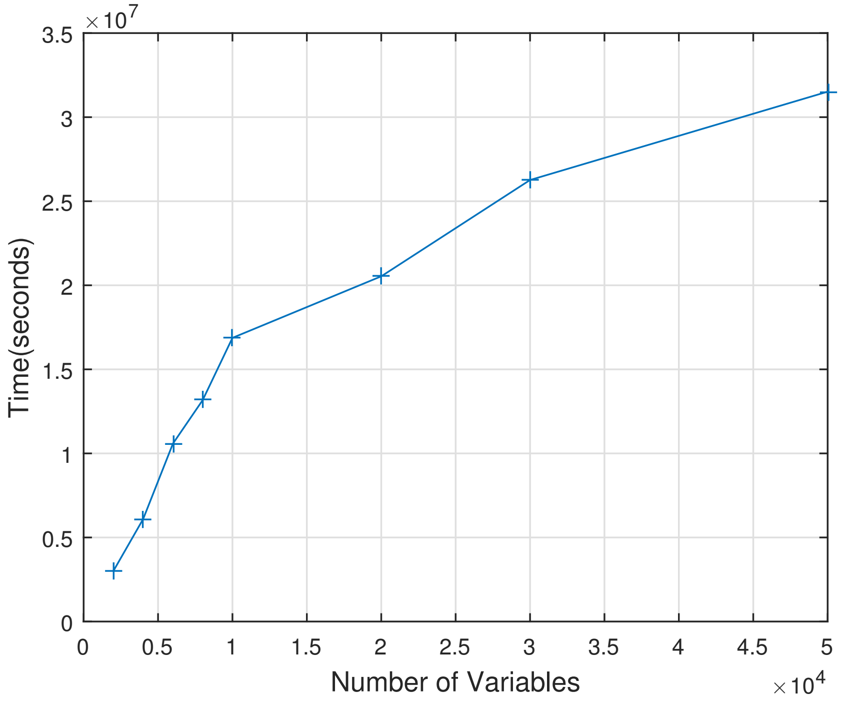

Figure 3.3. Computing time at the cloud of the proposed algorithm at the client and cloud for different separable program sizes.

3.7.2 Experiment Results

First, to explore the computing performance, we measure the computing time of our proposed algorithm both at the client and at the cloud, and show the results in Fig. 3.2 and Fig. 3.3, respectively. In particular, we measure the computation time of the client. That is, the time it takes to find the transformed problems, i.e., **P2** and **P3**, plus the time it takes to find the grid points. Fig. 3.2 shows the client's computing time for CSPs with an increasing number of variables. We observe that even when the CSP problem has a large number of variables, the client can still finish its local computations in a very short time. For example, the computing time of the client for the CSP with 2×10^4 variables is only 3.65×10^6s. For comparison, in Fig. 3.3, we show the total computing time at the cloud for solving the transformed CSPs with variables of different numbers. We observe a low computing time for the cloud even when the number of variables is very large. For instance, the cloud takes 5×10^8s to solve a CSP with 2×10^4 variables, which is very efficient in real-world scenarios.

Moreover, we summarize the total communication time of our algorithm under a 1Gbps link between the client and the cloud in Table 3.1. We observe that the communication time of our proposed algorithm is very small compared to the computing time. For example, the communication time of a CSP with size of 2×10^3 is 4.1×10^{-3}s, which is only about 0.5% of the total running time and can be neglected. Note that since commercial cloud computing services, such as Amazon Web Services, offer dedicated 1Gbps connections at a low price, e.g., \$0.3/hour. Therefore, it is a practical and cost-efficient for the client to employ high-speed links.

Table 3.1. Total Communication Time between the client and the cloud under a 1Gbps connection

CSP Size (variables)	**LP Size (bits)**	**Total Communication Time**
0.6×10^3	108.72×10^3b	1.23×10^{-3} s
1×10^3	181.2×10^3b	2.05×10^{-3}s
1.6×10^3	289.9×10^3b	3.28×10^{-3}s
2×10^3	362.4×10^6b	4.1×10^{-3}s

Next, we explore the computational savings offered by our proposed algorithm. Specifically, we compare the time it takes for the client to solve the CSPs by itself with that when the client and the cloud collaborate to solve the CSPs using our proposed secure outsourcing algorithm. We first show the time that the client takes to solve the CSPs with an increasing number of variables on its own in Fig. 3.4., We can see that it increases very fast. For example, the computing time of the client for solving a CSP with 2×10^4 variables is 3.86×10^8s, which is very inefficient for real-world applications. The reason behind this is that, solving large-scale CSPs requires a large amount of RAM and computing capacity, which is generally unavailable for clients such as small business companies and individuals. Without the help of cloud, the computation time increases exponentially as the number of variables goes up.

Furthermore, in order to demonstrate how much time we can save using the proposed algorithm for CSPs, we show the speedup offered by our proposed algorithm in Fig. 3.5. We calculate the speedup as the ratio between the time it takes the client to solve the CSPs by itself to its computing time under the proposed algorithm. We observe that our algorithm offers significant computing time savings to the client. For example, we observe that the speedup for a separable problem with 20×10^3 variables is $18.75\times$, that is, the client saves performing $18.75\times$ fewer operations, which is very impressive. Moreover, as the number of variables goes up, the speedup offered by our proposed algorithm becomes more and more significant, which means that with the integration of the cloud, our proposed algorithm saves more and more time for the client when the size of the problem increases. Therefore, we can say that the proposed algorithm not only protects the client's data privacy, but also significantly saves a lot of time for the client to solve large-scale CSPs.

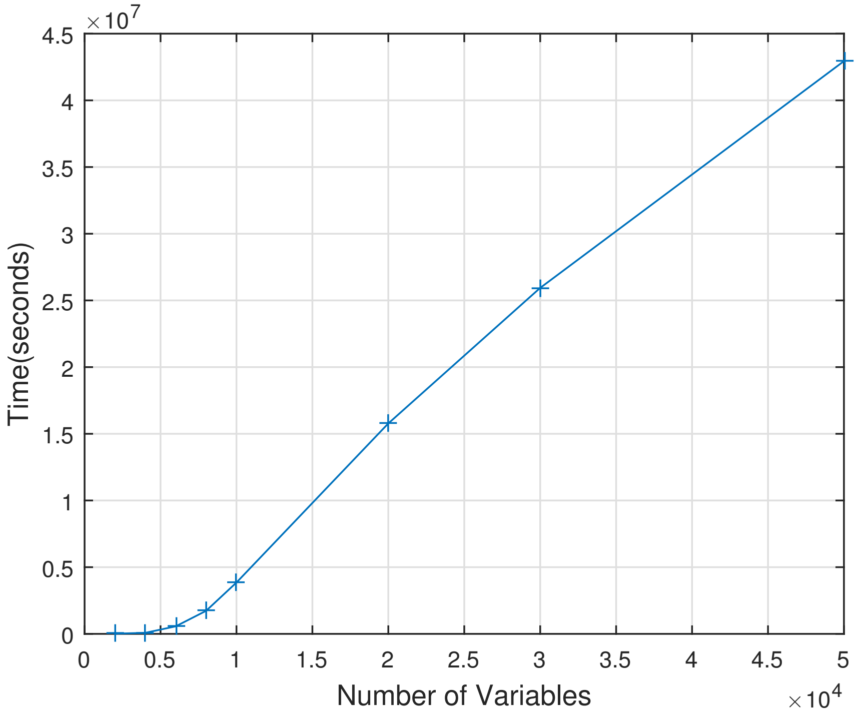

Figure 3.4. Computing time of the proposed algorithm at the client.

3.8 Conclusions

In this chapter, we have investigated the problem of secure outsourcing of large-scale CSPs. To the best of our knowledge, this is the first work to solve CSPs in a secure manner in cloud computing. To protect the client's private data, we have developed efficient vector and matrix transformation and permutation schemes that are solely based on linear algebra. We have shown that the values and positions of the transformed data are computationally indistinguishable from random vector and and matrices under chosen-plaintext attack (CPA), or CPA-secure. Therefore, the client can confidently share the transformed data with the cloud. The proposed secure linear approximation algorithm can enable the cloud server to efficiently find the solution while protecting the client's privacy. In addition, the correctness of the returned results from the cloud

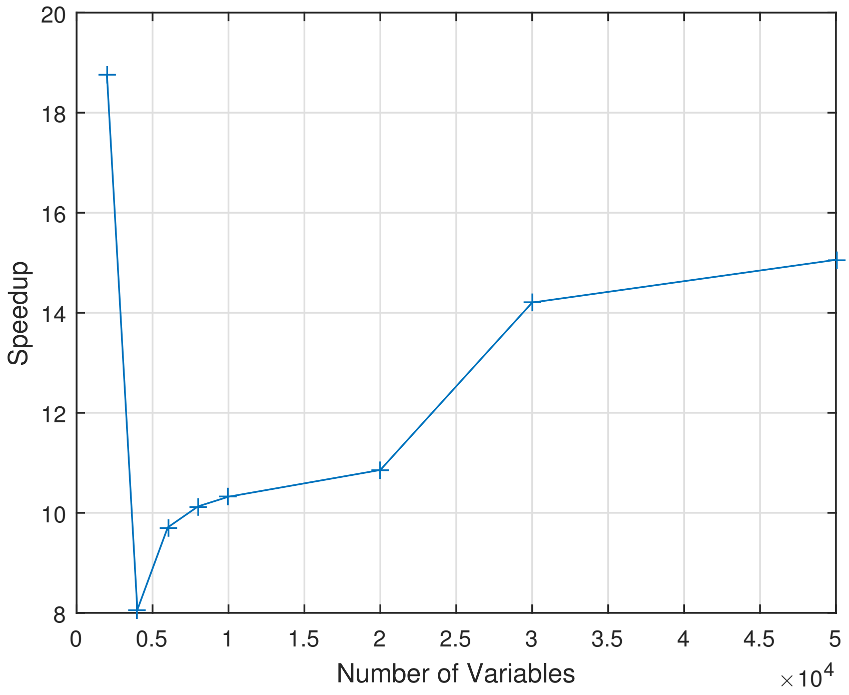

Figure 3.5. Computing speedup offered by the proposed algorithm.

can be efficiently verified by the client to prevent any malicious behavior of the cloud. The theoretical privacy analysis has demonstrated that the privacy of client's data is well preserved. Experimental results on Amazon Elastic Compute Cloud (EC2) have shown that the proposed algorithm can efficiently solve the large-scale CSPs with noticeable time savings for the client.

4 Conclusions and Future Work

4.1 Conclusions

The IoT technologies have brought both new features and significant security challenges to power systems. In this dissertation, we have investigated CFAs in power systems. Specifically, we have formulated a zero-sum stochastic game to analyze the interactions between an attacker and a system operator in dynamic environments for power systems. This problem is very complex and computationally intensive. Different from the previous work where complete enumeration of the system states is required, making the algorithms computationally intractable for large-scale power system applications, we pro- pose an efficient Q-CFA learning algorithm that only searches certain related possible actions for each player in the game, making the scheme scalable with fast convergence. We have also theoretically proven that the proposed algorithm achieves the Nash equilibrium. Moreover, considering that real-time statistics and sensitive data like system transition probabilities may not be accessible in practice, which unfortunately is an indispensable assumption in previous algorithms, our scheme works efficiently without requiring a priori knowledge of the system transition states. Simulation results show that by considering the system dynamics and the opponent's possible strategies, the optimal policy obtained by our proposed Q-CFA algorithm can achieve much better performance compared to several benchmark schemes.

Moreover, in this dissertation, we have also investigated the problem of secure outsourcing of large-scale CSPs. To the best of our knowledge, this is the first work to solve CSPs in a secure manner in cloud computing. To protect the client's private data, we have developed efficient vector and matrix transformation and permutation schemes that are solely based on linear algebra. We have shown that the values and positions of the transformed data are computationally indistinguishable from random vector and and matrices under chosen-plaintext attack (CPA), or CPA-secure. Therefore, the client can confidently share the transformed data with the cloud. The proposed secure linear approximation algorithm can enable the cloud server to efficiently find the solution while protecting the client's privacy. In addition, the correctness of the returned results from the cloud can be efficiently verified by the client to prevent any malicious behavior of the cloud. The theoretical privacy analysis has demonstrated that the privacy of client's data is well preserved. Experimental results on Amazon Elastic Compute Cloud (EC2) have shown that the proposed algorithm can efficiently solve the large-scale CSPs with noticeable time savings for the client. Moreover, our proposed scheme requires the client to help the cloud solve the problem, which incurs communication cost. In the future work, we plan to design a non-interactive secure outsourcing of CSPs scheme so that the client does not need to be involved during the process.

4.2 Future Work

My future research will focus on employing the big data techniques, system level security, decentralized system to construct intelligent and secure cyber physical systems and accommodate the smart city concept. In particular, apart from the cascading failure attack in cyber-physical systems, to further extend the presented work, we plan to keep working on this line of research and investigate further system level security problems in general IoT and CPS system with approaches game theory, data mining, and distributed computing. For example, cyber attack on state estimations in smart grids is an important research problem. State estimation is one of the most vital components in power grids. There have been several works studying the possibility and applicability of cyber attacks on state estimations in the smart grid. However, most of the works are based on DC power networks and can barely be applicable in real AC nonlinear power grids. We plan to develop the cyber attack-detection schemes and attack-defense schemes for state estimation in nonlinear smart grids.

Moreover, in the area of secure outsourcing of large-scale fundamental problems in the cloud, we will continue studying the secure outsourcing of large scale complex computations. For example, our proposed scheme requires the client to help the cloud solve the problem, which incurs communication cost. In the future work, we plan to design a non-interactive secure outsourcing of CSPs scheme so that the client does not need to be involved during the problem solving process. We will also study and design more efficient schemes for secure outsourcing of more complex problems such as exponential problems, general nonlinear problems, and so on. On the other hand, with the big data tools we have developed, we will also work on efficient big data application for smart city by taking advantage of secure outsourcing techniques that we have developed. For example, I will study secure cyber physical system by employing big data analytics to detect system attacks that bypass current detectors.

Complete References

[1] Siddhartha Kumar Khaitan and James D McCalley. Design techniques and applications of cyberphysical systems: A survey. IEEE Systems Journal, 9(2):350–365, 2015.

[2] Shichao Liu, Xiaoping P Liu, and Abdulmotaleb El Saddik. Denial-of-service (dos) attacks on load frequency control in smart grids. In IEEE PES Innovative Smart Grid Technologies (ISGT), pages 1–6, 2013.

[3] Yao Liu, Peng Ning, and Michael K Reiter. False data injection attacks against state estimation in electric power grids. ACM Trans. on Information and System Security, 14(1):13, 2011.

[4] Sergio Salinas, Changqing Luo, Weixian Liao, and Pan Li. State estimation for energy theft detection in microgrids. In Communications and Networking in China (CHINACOM), 2014 9th International Conference on, pages 96–101. IEEE, 2014.

[5] Aditya Ashok and Manimaran Govindarasu. Cyber attacks on power system state estimation through topology errors. In 2012 IEEE Power and Energy Society General Meeting, pages 1–8. IEEE, 2012.

[6] Mladen Kezunovic, Le Xie, and Santiago Grijalva. The role of big data in improving power system operation and protection. In Bulk Power System Dynamics and Control-IX Optimization, Security and Control of the Emerging Power Grid (IREP), 2013 IREP Symposium, pages 1–9. IEEE, 2013.

[7] Jie Chen, James S Thorp, and Ian Dobson. Cascading dynamics and mitigation assessment in power system disturbances via a hidden failure model. International Journal of Electrical Power & Energy Systems, 27(4):318–326, 2005.

[8] Weixian Liao, Sergio Salinas, Ming Li, Pan Li, and Kenneth A Loparo. Cascading failure attacks in the power system: a stochastic game perspective. IEEE Internet of Things Journal, 4(6):2247–2259, 2017.

[9] Ian Dobson, Benjamin A Carreras, Vickie E Lynch, and David E Newman. Complex systems analysis of series of blackouts: Cascading failure, critical points, and self-organization. Chaos: An Interdisciplinary Journal of Nonlinear Science, 17(2):026103, 2007.

[10] B Liscouski and W Elliot. Final report on the august 14, 2003 blackout in the united states and canada: Causes and recommendations. A report to US Department of Energy, 40(4), 2004.

[11] Mahshid Rahnamay-Naeini, Zhuoyao Wang, Nasir Ghani, Andrea Mammoli, and Majeed M Hayat. Stochastic analysis of cascading-failure dynamics in power grids. IEEE Trans. on Power Systems, 29(4):1767–1779, 2014.

[12] Jun Yan, Yufei Tang, Haibo He, and Yan Sun. Cascading failure analysis with dc power flow model and transient stability analysis. IEEE Trans. on Power Systems, 30:285–297, May 2014.

[13] Lanchao Liu, Mohammad Esmalifalak, Qifeng Ding, Valentine Emesih, and Zhu Han. Detecting false data injection attacks on power grid by sparse optimization. IEEE Trans. on Smart Grid, 5(2):612–621, Mar. 2014.

[14] ZJ Bao, YJ Cao, GZ Wang, and LJ Ding. Analysis of cascading failure in electric grid based on power flow entropy. Physics Letters A, 373(34):3032–3040, 2009.

[15] Ake J Holmgren, Erik Jenelius, and Jonas Westin. Evaluating strategies for defending electric power networks against antagonistic attacks. IEEE Trans. on Power Systems, 22(1):76–84, 2007.

[16] Javier Salmeron, Kevin Wood, and Ross Baldick. Analysis of electric grid security under terrorist threat. IEEE Trans. on Power Systems, 19:905–912, 2004.

[17] Guo Chen, Zhao Yang Dong, David J Hill, and Yu Sheng Xue. Exploring reliable strategies for defending power systems against targeted attacks. IEEE Trans. on Power Systems, 26(3):1000–1009, 2011.

[18] Nageswara SV Rao, Steve W Poole, Chris YT Ma, Fei He, Jun Zhuang, and David KY Yau. Cyber and physical information fusion for infrastructure protection: A game-theoretic approach. Technical report, Oak Ridge National Laboratory (ORNL), 2013.

[19] Chris YT Ma, David KY Yau, Xin Lou, and Nageswara SV Rao. Markov game analysis for attack-defense of power networks under possible misinformation. IEEE Trans. on Power Systems, 28(2):1676–1686, 2013.

[20] Oren Adaki. Attack on power lines leaves yemen in total darkness, June 2014.

[21] TES Raghavan and JA Filar. Algorithms for stochastic games, a survey. Zeitschrift für Operations Research, 35(6):437–472, 1991.

[22] Dimitri P. Bertsekas. Dynamic Programming and Optimal Control, volume 1 and 2. Athena Scientific, Belmont, Massachusetts, 2 edition, 2007.

[23] Javier Salmeron, Kevin Wood, and Ross Baldick. Worst-case interdiction analysis of large-scale electric power grids. IEEE Trans. on Power Systems, 24(1):96–104, January 2009.

[24] Yezhou Wang and Ross Baldick. Interdiction analysis of electric grids combining cascading outage and medium-term impacts. IEEE Trans. on Power Systems, PP(99):1–9, January 2014.

[25] Allen J Wood and Bruce F Wollenberg. Power generation, operation, and control. John Wiley & Sons, 2012.

[26] J. Thorp, A. Phadke, S. Horowitz, and S. Tamronglak. Anatomy of power system disturbances: importance sampling. International Journal of Electrical Power & Energy Systems, 20(2):147–152, 1998.

[27] Ian Dobson, Benjamin A Carreras, and David E Newman. A loading-dependent model of probabilistic cascading failure. Probability in the Engineering and Informational Sciences, 19(01):15–32, 2005.

[28] J. Chen, J. Thorp, and M. Parashar. Analysis of electric power system disturbance data. In Proceedings of the 34th Annual Hawaii International Conference on System Sciences, Washington, DC, USA, 2001.

[29] Xuan Liu, Kui Ren, Yanling Yuan, Zuyi Li, and Qian Wang. Optimal budget deployment strategy against power grid interdiction. In Proceedings of IEEE INFOCOM, Turin, Italy, April 2013.

[30] Abraham Neyman and Sylvain Sorin. Stochastic games and applications, volume 570. Springer, 2003.

[31] Jerzy Filar and Koos Vrieze. Competitive Markov decision processes. Springer-Verlag New York, Inc., 1996.

[32] S Tamronglak, SH Horowitz, AG Phadke, and JS Thorp. Anatomy of power system blackouts: preventive relaying strategies. IEEE Trans. on Power Delivery, 11(2):708–715, 1996.

[33] Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In Proceedings of the eleventh international conference on machine learning, volume 157, pages 157–163, 1994.

[34] Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 38(2):156–172, 2008.

[35] S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge University Press, 2004.

[36] John C Harsanyi and Reinhard Selten. A general theory of equilibrium selection in games. MIT Press Books, 1, 1988.

[37] G. Owen. Game Theory. Academic Press, 1982.

[38] Youngjune Gwon, Siamak Dastangoo, Carl Fossa, and HT Kung. Competing mobile network game: Embracing antijamming and jamming strategies with reinforcement learning. In Communications and Network Security (CNS), 2013 IEEE Conference on, pages 28–36. IEEE, 2013.

[39] Csaba Szepesvári and Michael L Littman. A unified analysis of value-function-based reinforcement-learning algorithms. Neural computation, 11(8):2017–2060, 1999.

[40] Junling Hu, Michael P Wellman, et al. Multiagent reinforcement learning: theoretical framework and an algorithm. In ICML, volume 98, pages 242–250. Citeseer, 1998.

[41] Ray Daniel Zimmerman, Carlos Edmundo Murillo-Sánchez, and Robert John Thomas. Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. IEEE Trans. on Power Systems, 26(1):12–19, 2011.

[42] IEEE 118 bus case flow limits. Illinois Institute of Technology.

[43] Vernon Turner, John F Gantz, David Reinsel, and Stephen Minton. The digital universe of opportunities: rich data and the increasing value of the internet of things. IDC Analyze the Future, Apr. 2014.

[44] Eric Bender. Big data in biomedicine. Nature, 527(7576):S1–S1, Apr. 2015.

[45] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. on knowledge and data engineering, 17(6):734–749, Jun. 2005.

[46] Ali Ipakchi and Farrokh Albuyeh. Grid of the future. IEEE Power and Energy Magazine, 7(2):52–62, Mar. 2009.

[47] Stefan M Stefanov. Separable programming: theory and methods. Springer Science & Business Media, Nov. 2013.

[48] Weixian Liao, Wei Du, Sergio Salinas, and Pan Li. Efficient privacy-preserving outsourcing of large-scale convex separable programming for smart cities. In IEEE 14th International Conference on Smart City. IEEE, Dec. 2016.

[49] Weixian Liao, Ming Li, Sergio Salinas, Pan Li, and Miao Pan. Energy-source-aware cost optimization for green cellular networks with strong stability. IEEE Trans. on Emerging Topics in Computing, 4(4):541–555, 2016.

[50] Weixian Liao, Ming Li, Sergio Salinas, Pan Li, and Miao Pan. Optimal energy cost for strongly stable multi-hop green cellular networks. In Distributed Computing Systems (ICDCS), 2014 IEEE 34th International Conference on, pages 62–72. IEEE, 2014.

[51] Dimitri P Bertsekas. Nonlinear programming. Athena scientific Belmont, Sep. 1999.

[52] Bao-Liang Lu and Koji Ito. Converting general nonlinear programming problems into separable programming problems with feedforward neural networks. Neural networks, 16(7):1059–1074, Sep. 2003.

[53] Weixian Liao, Changqing Luo, Sergio Salinas, and Pan Li. Efficient secure outsourcing of large-scale convex separable programming for big data. IEEE Transactions on Big Data, 2017.

[54] Ibrahim Abaker Targio Hashem, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, and Samee Ullah Khan. The rise of "big data" on cloud computing: Review and open research issues. Information Systems, 47:98–115, Jan. 2015.

[55] Will Venters and Edgar A Whitley. A critical review of cloud computing: researching desires and realities. Journal of Information Technology, 27(3):179–197, Sep. 2012.

[56] Sean Marston, Zhi Li, Subhajyoti Bandyopadhyay, Juheng Zhang, and Anand Ghalsasi. Cloud computing– the business perspective. Decision support systems, 51(1):176–189, Apr. 2011.

[57] Sheng Cai, Weixian Liao, Changqing Luo, Ming Li, Xiaoxia Huang, and Pan Li. Cril: An efficient online adaptive indoor localization system. IEEE Transactions on Vehicular Technology, 66(5):4148–4160, 2017.

[58] Ming Li, Weixian Liao, Xuhui Chen, Jinyuan Sun, Xiaoxia Huang, and Pan Li. Economic-robust transmission opportunity auction for d2d communications in cognitive mesh assisted cellular networks. IEEE Transactions on Mobile Computing, 2017.

[59] Cong Wang, Kui Ren, Jia Wang, and Karthik Mahendra Raje Urs. Harnessing the cloud for securely solving large-scale systems of linear equations. In 31st International Conference on Distributed Computing Systems (ICDCS). IEEE, Jun. 2011.

[60] Xinyu Lei, Xiaofeng Liao, Tingwen Huang, Huaqing Li, and Chunqiang Hu. Outsourcing large matrix inversion computation to a public cloud. IEEE Trans. on cloud computing, 1(1):78–87, 2013.

[61] Sergio Salinas, Changqing Luo, Xuhui Chen, Weixian Liao, and Pan Li. Efficient secure outsourcing of large-scale sparse linear systems of equations. IEEE Trans. on Big Data, 1(1):78–87, 2013.

[62] Nan Zhang and Wei Zhao. Privacy-preserving data mining systems. Computer, 40(4), 2007.

[63] Arun Thapa, Weixian Liao, Ming Li, Pan Li, and Jinyuan Sun. Spa: A secure and private auction framework for decentralized online social networks. IEEE Transactions on Parallel and Distributed Systems, 27(8):2394–2407, 2016.

[64] Cheng Sheng, Nan Zhang, Yufei Tao, and Xin Jin. Optimal algorithms for crawling a hidden database in the web. Proceedings of the VLDB Endowment, 5(11):1112–1123, 2012.

[65] Zheng Yan, Wenxiu Ding, Xixun Yu, Haiqi Zhu, and Robert H Deng. Deduplication on encrypted big data in cloud. IEEE Trans. on big data, 2(2):138–150, 2016.

[66] Tao Jiang, Xiaofeng Chen, and Jianfeng Ma. Public integrity auditing for shared dynamic cloud data with group user revocation. IEEE Trans. on Computers, 65(8):2363–2373, 2016.

[67] Sergio Salinas, Changqing Luo, Xuhui Chen, and Pan Li. Efficient secure outsourcing of large-scale linear systems of equations. In 2015 IEEE Conference on Computer Communications (INFOCOM). IEEE, Apr. 2015.

[68] Sergio Salinas, Changqing Luo, Weixian Liao, and Pan Li. Efficient secure outsourcing of large-scale quadratic programs. In Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, May 2016.

[69] Lifeng Zhou and Chunguang Li. Outsourcing large-scale quadratic programming to a public cloud. IEEE Access, 3:2581–2589, Dec. 2015.

[70] Xiaofeng Chen, Jin Li, Jianfeng Ma, Qiang Tang, and Wenjing Lou. New algorithms for secure outsourcing of modular exponentiations. IEEE Trans. on Parallel and Distributed Systems, 25(9):2386–2396, Sep. 2014.

[71] Zhan Qin, Jingbo Yan, Kui Ren, Chang Wen Chen, and Cong Wang. Towards efficient privacy-preserving image feature extraction in cloud computing. In Proceedings of the 22nd ACM international conference on Multimedia. ACM, Nov. 2014.

[72] Fei Chen, Tao Xiang, Xinyu Lei, and Jianyong Chen. Highly efficient linear regression outsourcing to a cloud. IEEE Trans. on Cloud Computing, 2(4):499–508, Dec. 2014.

[73] Arjun Dasgupta, Nan Zhang, Gautam Das, and Surajit Chaudhuri. Privacy preservation of aggregates in hidden databases: Why and how? In Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, pages 153–164. ACM, 2009.

[74] Chunqiang Hu, Abdulrahman Alhothaily, Arwa Alrawais, Xiuzhen Cheng, Carl Sturtivant, and Hang Liu. A secure and verifiable outsourcing scheme for matrix inverse computation. In INFOCOM 2017-IEEE Conference on Computer Communications, IEEE, pages 1–9. IEEE, 2017.

[75] Xinyu Lei, Xiaofeng Liao, Tingwen Huang, and Huaqing Li. Cloud computing service: The caseof large matrix determinant computation. IEEE Trans. on Services Computing, 8(5):688–700, Sep. 2015.

[76] Xinyu Lei, Xiaofeng Liao, Tingwen Huang, and Feno Heriniaina. Achieving security, robust cheating resistance, and high-efficiency for outsourcing large matrix multiplication computation to a malicious cloud. Information Sciences, 280:205–217, Oct. 2014.

[77] Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In Annual Cryptology Conference, pages 465–482. Springer, Aug. 2010.

[78] Fang Liu, Wee Keong Ng, and Wei Zhang. Encrypted gradient descent protocol for outsourced data mining. In 2015 IEEE 29th International Conference on Advanced Information Networking and Applications. IEEE, Mar. 2015.

[79] Mikhail J Atallah, Konstantinos N Pantazopoulos, John R Rice, and Eugene E Spafford. Secure outsourcing of scientific computations. Advances in Computers, 54:215–272, Dec. 2002.

[80] Cong Wang, Kui Ren, and Jia Wang. Secure and practical outsourcing of linear programming in cloud computing. In Proceeding of the IEEE International Conference on Computer Communications (INFOCOM'10), San Diego, California, USA, Apr. 2010.

[81] Cong Wang, Bingsheng Zhang, Kui Ren, and Janet M Roveda. Privacy-assured outsourcing of image reconstruction service in cloud. IEEE Trans. on Emerging Topics in Computing, 1(1):166–177, Jun. 2013.

[82] Mokhtar S Bazaraa, Hanif D Sherali, and Chitharanjan M Shetty. Nonlinear programming: theory and algorithms. John Wiley & Sons, jun. 2013.

[83] Jonathan Katz and Yehuda Lindell. Introduction to modern cryptography. CRC press, Nov. 2014.

[84] Masakazu Kojima, Shinji Mizuno, and Akiko Yoshise. A primal-dual interior point algorithm for linear programming. In Progress in mathematical programming, pages 29–47. Springer, 1989.

[85] Cong Wang, Kui Ren, and Jia Wang. Secure optimization computation outsourcing in cloud computing: A case study of linear programming. IEEE Trans. on Computers, 65(1):216–229, Jan. 2016.

[86] Amazon. What is amazon ec2. https://aws.amazon.com/ec2/.