ABSTRACT

JIN, RICHENG. Privacy-Preserving Information Exchange in Collaborative Security, Crowdsensing, and Machine Learning. (Under the direction of Dr. Huaiyu Dai).

The explosive development in modern technologies brings unprecedented opportunities in collaboration among network entities through information exchange. In the meantime, we have witnessed the emergence of various security and privacy problems. In this dissertation, we explore the security and privacy problems in three important emerging areas: collaborative security, crowdsensing, and machine learning.

The first part of this dissertation focuses on collaborative security. To cope with the increasingly sophisticated intrusions, collaborative intrusion detection systems (CIDSs) are proposed in the literature. In particular, intrusion detection systems (IDSs) in collaboration can dynamically share available computational resources with each other to enhance the overall detection performance. However, due to resource limitation, it is infeasible for the IDSs to respond to all the intrusion detection requests from their collaborative peers. In the meantime, obtaining the optimal IDS configuration in CIDSs is far from trivial. With such consideration, in Chapter 2, the collaborative IDS configuration problem is formulated as a two-layer stochastic game (SG). To solve the two-layer SG, a centralized Vickrey–Clarke–Groves (VCG) auction based collaboration scheme and a distributed game-theoretic incentive mechanism are proposed. It is shown that the proposed IDS collaboration schemes can achieve both efficient configuration and effective collaboration. Considering that sharing the security-related information in collaborative security may lead to privacy concerns, the security-privacy tradeoff is investigated in Chapter 3. The interplay between the attacker and the collaborative security entities (e.g., IDSs) is formulated as Quantitative Information Flow (QIF) games, in which the QIF theory is adapted to measure the collaboration gain and the privacy loss of the entities in the information sharing process. More specifically, three QIF games are formulated and solved, each corresponding to one possible scenario of interest in practice. Based on the game-theoretic analyses, the expected behaviors of both the attacker and the security entities are obtained. Through numerical computation, we obtain the security-privacy tradeoff curves, with which the collaborative entities can better evaluate the privacy cost of achieving certain security performance.

Chapter 4 is devoted to location privacy preservation in crowdsensing, in which multiple mobile agents are employed by a Base Station (BS) to perform location-dependent information collection tasks. In this case, the mobile agents are required to share their locations with the BS, which incurs location privacy concerns and may deter them from participating in the information delivery process. With such consideration, a location privacy-aware incentive mechanism is proposed for the BS to incentivize the mobile agents to trade their location privacy with the BS.

Different from most of the existing approaches that assume fixed privacy levels, the proposed incentive mechanism allows the BS to motivate the mobile agents to report their locations with differential privacy levels desired by the BS. Furthermore, considering that the BS usually has a limited budget, it is essential to properly select the set of mobile agents to perform the information collection tasks. Therefore, a cost-efficient mobile agent selection algorithm is proposed. The effectiveness of the proposed incentive mechanism and the mobile agent selection algorithm is demonstrated through numerical simulations.

Chapter 5 and Chapter 6 are dedicated to the machine learning field. Particularly, we focus on federated learning (FL), which is a prominent distributed learning paradigm that enhances user privacy since the decentralized data reside on mobile devices during the training process. However, FL entails some pressing needs for developing novel parameter estimation approaches with theoretical guarantees of convergence, which are also communication efficient, differentially private, and Byzantine resilient in the heterogeneous data distribution settings. To improve the communication efficiency, Quantization-based SGD solvers have been widely adopted in FL and the recently proposed SIGNSGD with majority vote shows a promising direction. Nonetheless, no existing methods enjoy all the aforementioned properties. In Chapter 5, we propose Stochastic-Sign SGD, which utilizes novel stochastic-sign based gradient compressors to enable the aforementioned properties in a unified framework. We also present an error-feedback variant of the proposed Stochastic-Sign SGD which further improves the learning performance in FL. The practical implementation of the aforementioned sign based SGD algorithms over wireless networks is investigated in Chapter 6. Different from most of the existing works that consider Channel State Information (CSI) at both the transmitter and the receiver sides, only receiver side CSI is assumed in our study. In such a case, an essential problem for the mobile devices is to select appropriate local processing and communication parameters to achieve the desired balance between the overall learning performance and their energy consumption. For SIGNSGD, two optimization problems are formulated and solved. The first problem minimizes the energy consumption of the workers given the learning performance requirement, while the second problem optimizes the learning performance given the energy consumption requirement. Furthermore, considering that the data are distributed across the mobile devices in a highly uneven fashion in FL, Stochastic-Sign SGD is adapted by considering the outage probability of the workers. Compared with SIGNSGD, the proposed stochastic-sign based algorithm achieves better learning performance while reducing the energy consumption of the mobile devices.

In summary, several interesting and important security and privacy problems in the three aforementioned emerging areas have been investigated in this dissertation. We hope that our work can stimulate further development in these burgeoning research fields.

Privacy-Preserving Information Exchange in Collaborative Security, Crowdsensing, and Machine Learning

by
Richeng Jin

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Electrical Engineering

Raleigh, North Carolina

2020

APPROVED BY:

Dr. Brian Hughes	Dr. Dror Baron
Dr. Wenye Wang	Dr. Huaiyu Dai Chair of Advisory Committee

DEDICATION

To my family.

BIOGRAPHY

Richeng Jin received his Bachelor of Engineering degree in Information and Communication Engineering in 2015 from Zhejiang University, Hangzhou, China. In August 2015, he started his work towards the Ph.D. degree in Electrical Engineering at North Carolina State University, Raleigh, NC, under the guidance of Prof. Huaiyu Dai. His doctoral research interests mainly focus on network security and machine learning.

ACKNOWLEDGEMENTS

First of all, I would like to express my deepest gratitude to my advisor, Professor Huaiyu Dai, for his patient guidance and continuous support throughout my doctoral research journey. Besides, I am extremely grateful for the invaluable advice and encouragement in my life and career development. It is a great honor for me to work with him and learn from his integrity, humility, and kindness.

My appreciation also goes to Professor Brian Hughes, Professor Dror Baron, Professor Wenye Wang, Professor Min Kang, and Professor Chau-Wai Wong, for their enlightening instructions and feedback. I also want to say a big thank you to all the professors and staff who have provided assistance in the completion of my doctoral study.

Furthermore, I am grateful to my friends and colleagues at NC State, Xiaofan He, Yufan Huang, Juan He, Srinjoy Chattopadhyay, Seyyedali Seyyedali, Ali Rahmati, Anuj Nayak, Junan Zhu, Xian Yang, and Yang Shi. Their friendship will be a great treasure for the rest of my life.

Besides, I would like to acknowledge the financial support from the US National Science Foundation (under grants ECCS-1444009 and CNS-1824518) and the Army Research Office (under Grant W911NF-17-1-0087).

Finally, I wish to say a special thank you to my dear parents. I never thank you enough for encouraging me to pursue my dream and guiding me in the right direction. Without their love and support, I could not be the person I am today.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

The advances in communication technologies enable individuals and devices to communicate with each other at an extremely low cost and therefore play an increasingly important role in modern society. In collaborative security, the security entities (e.g., intrusion detection systems) can benefit from the knowledge about the attacks shared by other peers, which enhances the overall intrusion detection accuracy. In crowdsensing, mobile agents can collect and share information to help the platform finish the sensing tasks in a cost-effective way. In machine learning, the users can collect local training data and collaboratively train a machine learning model in a distributed and efficient manner. While bringing unprecedented opportunities in these areas, the information exchange introduces new security and privacy challenges. In this dissertation, we focus on studying the collaboration schemes in collaborative security and the tradeoff between security and privacy, developing location privacy-preserving crowdsensing schemes, designing and implementing the communication-efficient federated learning algorithms over wireless networks.

1.1 Privacy-aware Information Exchange in Collaborative Security

We start our study from the intrusion detection system (IDS) collaboration problem in collaborative security. As an important defense mechanism against various network intrusions, the IDSs monitor the network status and detect suspicious activities. With the rapid development of attacking tactics, detecting the increasingly sophisticated intrusions is becoming much harder for individual IDSs. To mitigate this problem, intrusion detection networks (IDNs) which consist of multiple collaborative IDSs have been developed in the literature [2–4]. In a consultation based IDN, when an IDS does not have sufficient confidence to make a decision, it may send consultation requests to other more powerful IDSs and ask them to help detect intrusions [5, 6]. On the one hand, due to resource limitation, it is impossible for the IDSs to respond to all the

requests from their collaborative peers, and hence an adequate strategy is needed to achieve efficient collaboration. On the other hand, the collaboration procedure makes it highly non-trivial for the IDSs to obtain their optimal configuration. In Chapter 2, the collaborative IDS configuration problem is modeled as a two-layer stochastic game, in which the first layer deals with the interaction between each IDS and the corresponding attacker and the second layer deals with the collaboration among IDSs. Two IDS collaboration schemes, which are among the first approaches that address the IDS collaboration problem and the IDS configuration problem simultaneously, are proposed for the centralized and the distributed settings, respectively.

Despite that sharing the security-related information enhances the overall detection accuracy, it leads to potential privacy leakage, which may prevent the security entities to participate in the collaboration, especially when they are self-interested. Although various privacy-preserving techniques have been proposed for collaborative defense in the literature [7–15], they often lead to a certain level of utility degradation. In addition, it is often difficult to quantify the amount of preserved privacy and utility loss in the existing methods. Moreover, the existing methods do not have the flexibility of properly adjusting the collaboration strategies in response to a given privacy requirement. In Chapter 3, the tradeoff between security and privacy in collaborative security is studied. To quantitatively measure the amount of preserved privacy and the corresponding utility loss, the Quantitative Information Flow (QIF) theory, which has seen rapid development in the past decade [16–18], is adopted. Three QIF games are formulated, each corresponding to one possible scenario of interest in practice. By solving the QIF games, the optimal attacking strategy for the attacker, and the optimal collaboration strategies for the entities in different settings are obtained.

1.2 Location Privacy Preservation in Crowdsensing

Many emerging applications, for instance, traffic monitoring [19], noise pollution monitoring [20] and air quality monitoring [21], depend on the collection of status updates from a set of ground terminals that are equipped with specialized sensors. On the one hand, these applications require timely status updates for information freshness. The recently proposed metric Age of Information (AoI) [22] can capture the timeliness of the updates in these applications. On the other hand, considering that the ground terminals may not be able to communicate with the Base Station (BS) directly due to power limitation [23], mobile agents are commonly employed to help gather the information updates [23–25]. In particular, mobile crowd sensing (MCS) based methods have achieved remarkable performance in numerous time-sensitive applications [26–28]. In an MCS system, a number of mobile agents, equipped with mobile devices, are employed to perform sensing tasks. Since the information collection task is location-sensitive, at each time slot, the mobile agents are supposed to report their locations before the BS determines whether to select

and pay them to help deliver information or not. However, reporting the locations may incur location privacy concerns for the mobile agents. Therefore, a location privacy-aware incentive mechanism is in need to motivate the mobile agents to participate in the information delivery process.

Protecting location privacy is essential when location privacy-aware mobile agents are considered, and therefore, has attracted lots of research interests. Conventional approaches regarding location privacy preservation include cloaking [29] and k-anonymity [30]. However, such approaches are vulnerable to adversaries with prior knowledge about the locations of the mobile agents [31]. To address this problem, differential privacy [32] has been introduced for location privacy preservation [31, 33–38]. Nonetheless, most of the existing approaches only use differential privacy as a tool for location privacy protection and assume a fixed privacy level. [39] considers differential location privacy in the design of the incentive mechanism. However, it assumes that the workers share their true location information with a trustworthy BS which releases the location information in a differentially private manner. In this sense, the workers do not have control over their own location information. A trading market is developed in [40] which allows the workers to determine their own privacy levels and trade their location privacy. However, since auction-based methods are used, the workers are supposed to determine their privacy levels before submitting their bids. Different from these works, in Chapter 4, the location differential privacy is considered as a "commodity" that the mobile agents are willing to trade, together with their working efforts, with the BS for a higher payment. An incentive mechanism is proposed for the BS to incentivize the mobile agents to trade their location privacy with the BS. Compared to the existing methods in the literature, our proposed incentive mechanism allows the BS to adjust its payment mechanism to motivate the mobile agents to select its desired privacy levels. Given the incentive mechanism, a cost-efficient mobile agent selection algorithm is proposed to minimize the AoI of the ground terminals.

1.3 Privacy-Preserving Federated Learning over Wireless Networks

Besides collaborative security and crowdsensing, machine learning is another emerging area that benefits from efficient information exchange. To train a machine learning model, traditionally a centralized approach is adopted in which the training data are aggregated on a single machine. Such a centralized training approach is privacy-intrusive, especially when the data are collected by mobile devices and contain the owners' sensitive information (e.g., locations, user preference on websites, social media). With such consideration, Federated Learning (FL) has become a prominent distributed machine learning paradigm since it allows training on a large amount

of decentralized data residing on devices like mobile phones [1] and therefore provides certain privacy guarantees for the mobile devices. During each communication round, after receiving the learning model parameters from the server, the workers (e.g., mobile devices) train their local learning models using their local data and transmit the parameter updates back to the server, which will aggregate the information from all the workers and start the next round by broadcasting the updated model parameters. FL imposes several critical challenges. First of all, the communication capability of mobile devices can be a significant bottleneck. Furthermore, the training data on a given worker is typically based on its usage of the mobile devices, which results in heterogeneous data distribution. Finally, the local data usually contain some sensitive information about a particular mobile device user. Despite that the training data are kept private on the mobile devices, adversaries may still infer sensitive information from the parameter updates shared by the workers [41]. Therefore, there is a pressing need to develop a privacy-preserving distributed learning algorithm. Finally, similar to many distributed learning methods, FL may suffer from malicious participants. As is shown in [42], even a single Byzantine worker, which may transmit arbitrary information, can severely disrupt the convergence of distributed gradient descent algorithms. However, to the best of our knowledge, no existing methods can cope with all the aforementioned challenges.

To alleviate the communication burden of the workers, there have been various gradient quantization methods [43–47] in the literature, among which the recently proposed SIGNSGD with majority vote [48] is of particular interest due to its robustness and communication efficiency.[1] In SIGNSGD, during each communication round, only the signs of the gradients and aggregation results are exchanged between the workers and the server, which leads to around $32\times$ less communication than full-precision distributed stochastic gradient descent (SGD). Nonetheless, it has been shown in [49] that SIGNSGD fails to converge when the data on different workers are heterogeneous (i.e., drawn from different distributions), which is one of the most important features in FL. In Chapter 5, we propose Stochastic-Sign SGD, which is a class of stochastic-sign based SGD algorithms and guarantees to converge in the heterogeneous data distribution setting. More specifically, two compressors *sto-sign* and *dp-sign* are proposed, which extend SIGNSGD to its stochastic and differentially private variants. In order to further improve the learning performance, the error-feedback technique is incorporated, which extends the proposed algorithms to their error-feedback variants.

In Chapter 6, the practical implementation of the sign based SGD algorithms over wireless networks is studied. Since all the communications between the workers and the server are over wireless links, the learning performance depends on the wireless environments as well as the workers' communication resource and energy constraints. There have been some works that

[1]Note that all the algorithms considered in this chapter use the idea of majority vote. Therefore, we ignore the term "with majority vote" in the following discussions for ease of presentation.

study the communication aspects of FL [50–62]. Nonetheless, they either do not consider the existing strategies that have shown promising improvement in communication efficiency (e.g., gradient quantization [63]) or ignore the energy consumption of the workers, the impact of transmission errors, and data heterogeneity. In addition, all these works assume channel state information (CSI) at both the server side and the worker side, which may not be reasonable in practice. In Chapter 6, the workers are assumed to transmit their parameter updates over flat-fading channels and CSI is only available at the receiver side. Channel capacity with outage is considered and each worker is supposed to determine its transmission rate and transmission power. It is worth mentioning that in real-world FL applications over wireless networks, the communication time between the server and the workers is not negligible. Therefore, it becomes more critical to improve the learning performance with respect to the total training time instead of the number of communication rounds. With such consideration, the implementation of the FL algorithms given a fixed total training time is considered. In such a case, the learning performance depends on the number of communication rounds that the FL algorithm can be run and the outage probabilities of the workers for each communication round. In addition, considering that mobile devices usually have limited batteries, it is essential to minimize their energy consumption by appropriately configuring the local computation and communication parameters while satisfying the learning performance requirement (or the other way around). In Chapter 6, in the homogeneous data distribution scenario, two optimization problems are formulated and solved. The first problem minimizes the energy consumption of the workers given the learning performance requirement, while the second problem optimizes the learning performance given the energy consumption requirement. Furthermore, in the heterogeneous data distribution scenario, Stochastic-Sign SGD is adapted by incorporating the outage probability of the workers, which outperforms SIGNSGD in learning performance while reducing the energy consumption of the mobile devices.

1.4 Organizations

The remainder of this dissertation is organized as follows. Two collaborative IDS schemes are proposed in Chapter 2. The tradeoff between security and privacy in collaborative security is investigated in Chapter 3. The location privacy-preserving crowdsensing scheme is proposed in Chapter 4. The communication efficient federated learning algorithms and their implementation over wireless networks are presented in Chapter 5 and Chapter 6, respectively. Finally, the dissertation is concluded in Chapter 7, together with some possible directions for future work.

Chapter 2

Collaborative IDS Configuration: A Two-Layer Game-Theoretic Approach

In this chapter, two IDS collaboration schemes that can achieve both efficient configuration and effective collaboration are proposed for the centralized and the distributed settings, respectively. In the considered framework, the IDSs with ample computational resources (e.g., CPU time and memory) are allowed to help run intrusion detection algorithms for the IDSs with insufficient resources. As a result, the overall detection rate of the IDN can be enhanced. To find an effective collaboration strategy for such a dynamic IDS collaboration problem, a two-layer stochastic game (SG) is proposed in this chapter. Specifically, the first layer deals with the interaction between each IDS and the corresponding attacker. In the interaction, both the IDSs and the attackers can use learning algorithms (e.g., Nash Q-learning [64]) to gradually learn their own strategies. The second layer deals with the collaboration among IDSs in the IDN. To promote collaboration among IDSs, a Vickrey-Clarke-Groves (VCG) auction [65] based collaboration scheme is proposed. In this scheme, an IDN manager that can help determine a resource allocation strategy to optimize the overall detection performance of the entire IDN is assumed available. When such an IDN manager is not available, a distributed game-theoretic incentive mechanism is further developed. In addition, the proposed collaboration schemes are suitable for general collaborative security problems and hence can find broader applications beyond collaborative intrusion detection.

The remainder of this chapter is organized as follows. Section 2.1 formulates the collaborative IDS configuration problem. The VCG auction based two-layer collaboration scheme is presented in Section 2.2. The distributed incentive mechanism for IDS collaboration is presented in Section 2.3. The effectiveness of the proposed algorithms is examined through simulations in

Section 2.4. Related works are discussed in Section 2.5. Section 2.6 concludes this chapter.

2.1 Problem Formulation

In this section, a two-layer SG is formulated for the collaborative IDS configuration problem. Important notations used in this chapter are summarized in Table 2.1. An IDN that consists of N IDSs is considered, denoted by $\mathcal{N} = \{1, 2, ..., N\}$. Without loss of generality, the following assumptions are made:

- The IDSs in the IDN are host-based and signature-based; each of them monitors the network activities of one subsystem in the IDN.
- Each IDS faces one attacker that can launch multiple attacks simultaneously.
- Each subsystem has two possible states $\{H, I\}$, where H (I) stands for the healthy (ill) state. Here the healthy state refers to the situation that the subsystem is in a normal operating condition while the ill state refers to the case that the subsystem is working in a vulnerable state. However, the subsystem can recover from the ill state once the intrusion is detected and the vulnerabilities are fixed.[1] Take the CPU exhaustion attacks [66] as an example. Specifically, the attacker can launch the flood attacks, and if successful, the resource allocated to the IDSs will be greatly suppressed due to CPU exhaustion at the host. In this case, the ill state can refer to a state that the adversary has successfully conducted a DoS attack and depleted a significant portion of the computational resource (e.g., CPU) of the host. The memory exhaustion attacks [66] induce similar effects. In addition, it is assumed that the IDSs can monitor the states of the subsystems.[2]
- Host-based IDSs are assumed in this chapter, and hence they consume the computational resources (e.g., CPU) of the subsystems they are monitoring [67]. As a result, more computing resources can be allocated to an IDS when the corresponding subsystem is in the healthy state. In contrast, when a subsystem is in the ill state (due to previously undetected attacks), some computational resources may be compromised [68], and therefore, fewer resources will be available to support the associated IDS. For CPU exhaustion attacks and signature-based IDSs, the intrusion detection rates of the IDSs may substantially degrade since fewer signatures can be compared in each timeslot due to CPU resource shortage. In this chapter, the computational resource difference of different system states is measured by the number of signature libraries that the IDSs can load and utilize for real-time comparison based on the current computational resources.

[1] A similar model can be found in [64].

[2] Since each IDS corresponds to one subsystem, whenever the state of an IDS is mentioned, we are referring to the state of the subsystem that the IDS is monitoring, for ease of presentation.

Table 2.1: Important notations

$\mathcal{N}$	the set of IDSs.
s^i	the state of subsystem i.
$U_i(s^i)$	the number of libraries IDS_i can load at state s^i
$\mathcal{L}$	the set of detection libraries.
$\mathcal{A}$	the set of attacks.
$\boldsymbol{l}^i$	the configuration of IDS_i.
$\boldsymbol{a}^i$	the action of attacker γ_i.
p_{l_n,a_m}	the probability of attack a_m been detected by library l_n.
p_{st}	state transition probability.
$w^{IDS_i}_{s^i,a_k}$	the importance of detecting attack a_k for IDS_i at state s^i.
$w^{\gamma_i}_{s^i,a_k}$	the profit of fulfilling attack a_k for attacker γ_i at state s^i.
g_{ij}	the amount of resource that IDS_i shares IDS_j.
π^{IDS}	the strategy of IDS.
π^{γ}	the strategy of attacker.
$\tilde{\pi}^{\gamma}$	the attacker's strategy estimated by IDS.

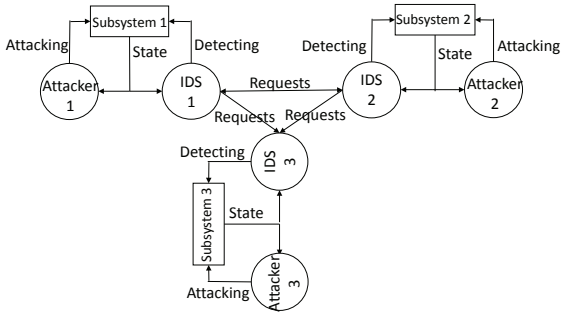

Figure 2.1: Block diagram of an IDN.

- The action sets of both the IDSs and the attackers are assumed fixed and publicly known [64] in this study, while unknown attacks are left to future work.

Fig. 2.1 depicts the scenario of three fully connected IDSs facing three independent attackers. Let $\mathcal{S} = \{s^1, s^2, ..., s^N, U_1(s^1), U_2(s^2), ...U_N(s^N)\}$ denote the system state, in which $s^i \in \{H, I\}$ represents the state of subsystem i, and $U_i(s^i)$ represents the number of libraries IDS_i can load depending on the current state of subsystem i. Let $\mathcal{L} = \{l_1, l_2, ..., l_L\}$ denote the set of detection libraries available to all the IDSs. Denoted by $\mathcal{L}^* = \sigma(\mathcal{L})$ the power set of $\mathcal{L}$, with cardinality $|\mathcal{L}^*| = 2^L$. Each $\boldsymbol{l}^i \in \mathcal{L}^*$ corresponds to a possible configuration of IDS_i.

It is assumed that the attackers (denoted by γ) can launch L different attacks $\mathcal{A} = \{a_1, a_2, ..., a_L\}$ and attack a_m will be detected by library l_n with probability p_{l_n,a_m}. The detection probability p_{l_n,a_m} is assumed high when the library matches the attack (i.e., $m = n$) and low

otherwise. Similarly, each attacker γ_i can launch a set of attacks $\boldsymbol{a}^i \in \mathcal{A}^*$, with $\mathcal{A}^* = \sigma(\mathcal{A})$ the power set of $\mathcal{A}$.

To model the influence of library configuration $\boldsymbol{l}^i$ and attacks $\boldsymbol{a}^i$ on the subsystem state, it is assumed that the subsystem state will transit from s^i to $s^{i'}$, with probability $p_{st}(s^{i'}|s^i, \boldsymbol{l}^i, \boldsymbol{a}^i)$. When the subsystem is at state s^i and IDS_i loads the set of libraries $\boldsymbol{l}^i$ while the attacker γ_i takes the set of attacks $\boldsymbol{a}^i$, the reward function of IDS_i is modeled as

$$R^{IDS_i}(s^i, \boldsymbol{l}^i, \boldsymbol{a}^i) = \sum_{a_k \in \boldsymbol{a}^i} p_{\boldsymbol{l}^i, a_k} w^{IDS_i}_{s^i, a_k}, \tag{2.1}$$

where $w^{IDS_i}_{s^i, a_k}$ represents the importance of detecting attack a_k when subsystem i is at state s^i, and $p_{\boldsymbol{l}^i, a_k}$ refers to the probability of detecting a_k when IDS_i loads the set of libraries $\boldsymbol{l}^i$, which is given by

$$p_{\boldsymbol{l}^i, a_k} = 1 - \prod_{l_n \in \boldsymbol{l}^i} (1 - p_{l_n, a_k}). \tag{2.2}$$

Note that the modeling in (2.1) conforms to the intuition that the IDS can obtain a better reward when it loads the right library. The term w^{IDS}_{s, a_k} is motivated by the fact that the damage caused by the same attack at different system states may be different. In this chapter, homogeneous IDSs and attackers are considered.

The reward function of the attacker γ_i is modeled as

$$R^{\gamma_i}(s^i, \boldsymbol{l}^i, \boldsymbol{a}^i) = \sum_{a_k \in \boldsymbol{a}^i} (1 - p_{\boldsymbol{l}^i, a_k}) w^{\gamma_i}_{s^i, a_k}, \tag{2.3}$$

where $w^{\gamma_i}_{s^i, a_k}$ denotes the profit of fulfilling attack a_k for attacker γ_i when subsystem i is at state s^i. To maximize its reward, IDS_i has to form a strategy $\pi^{IDS_i}(s^i, \boldsymbol{l}^i)$ to select its detection libraries that match the potential attacks from the attacker γ_i.

In the considered collaborative IDS configuration problem, the connected IDSs are allowed to cooperate through resource sharing. Let g_{ij} be the amount of resource that IDS_i offers to IDS_j, for $i, j \in \mathcal{N}$. The resource that can be shared by each IDS is restricted by its state-dependent capacity, i.e.,

$$\sum_{j \in \mathcal{N}} g_{ij} \leqslant U_i(s^i), \forall i \in \mathcal{N}. \tag{2.4}$$

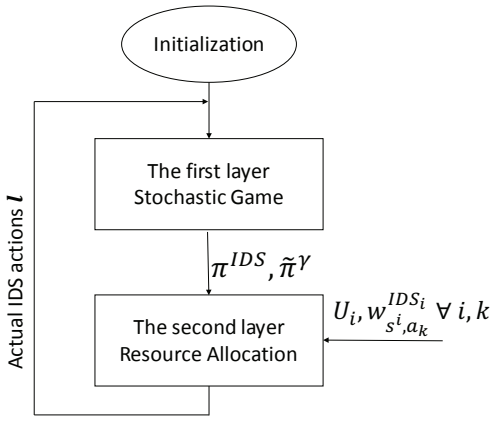

Figure 2.2: Overview of the proposed approach.

2.2 VCG Auction Based Collaboration Scheme

The VCG auction based collaboration scheme for collaborative IDS configuration is illustrated in this section. In each round of the proposed scheme, the IDSs will first interact with their corresponding attackers to learn their strategies, and then all the IDSs will report to the IDN manager the amount of resources they have as well as their expected reward functions.[3] The IDN manager then determines a resource allocation scheme for a better overall detection performance. Fig. 2.2 gives an overview of this scheme.

2.2.1 First Layer: Stochastic Game

The first layer is concerned with the interaction between each IDS and its corresponding attacker. The objective of each IDS (attacker) is to maximize its cumulative discounted reward $\mathbb{E}\{\sum_{n=1}^{\infty} \beta^n R_n^{IDS}\}$ ($\mathbb{E}\{\sum_{n=1}^{\infty} \beta^n R_n^{\gamma}\}$) with discounting factor $\beta \in [0,1)$ representing its long-term performance with diminishing weight on the future. The IDS (attacker) needs to learn a strategy $\pi^{IDS}(s, \boldsymbol{l})$ ($\pi^{\gamma}(s, \boldsymbol{a})$) which specifies the probability of taking action $\boldsymbol{l} \in \mathcal{L}^*$ ($\boldsymbol{a} \in \mathcal{A}^*$) at a given state s. To this end, the interaction can be formulated as a SG as follows: the IDS and the attacker are the two players; the set of possible subsystem states defines the state space of the stochastic game; $\sigma(\mathcal{A})$ and $\sigma(\mathcal{L})$ are the action space of the IDS and the attacker; and the state transition function $p_{st}(s'|s, \boldsymbol{l}, \boldsymbol{a})$ defines the probability of reaching a future state given the current state and actions of both the IDS and the attacker. Interested readers may refer to [69] for more details on SG.

[3]It is assumed that the IDN manager is secure and able to verify the messages from IDSs and thus no IDS will send false information. Selfish and malicious IDSs will be considered in our future work.

The Nash-Q learning algorithm [69] can be employed to solve the SG described above. At each time slot n, after observing the actions $(\boldsymbol{l}_n, \boldsymbol{a}_n)$, the reward $r_n^{IDS} = R^{IDS}(s_n, \boldsymbol{l}_n, \boldsymbol{a}_n)$, and the state transition from s_n to s_{n+1}, the IDS updates the quality and the value functions Q^{IDS} and V^{IDS} for itself. It will also maintain a pair of virtual quality functions $\tilde{Q}^{\gamma}$ and $\tilde{V}^{\gamma}$, to keep track of the attacker's behavior. In particular, these quantities are updated as follows:

$$Q_{n+1}^{IDS}(s, \boldsymbol{l}, \boldsymbol{a}) \tag{2.5}$$
$$= \begin{cases} (1-\alpha_n) Q_n^{IDS}(s, \boldsymbol{l}, \boldsymbol{a}) + \alpha_n \left[r_n^{IDS} + \beta \cdot V_n^{IDS}(s_{n+1}) \right], & \text{for } (s, \boldsymbol{l}, \boldsymbol{a}) = (s_n, \boldsymbol{l_n}, \boldsymbol{a_n}), \\ Q_n^{IDS}(s, \boldsymbol{l}, \boldsymbol{a}), & \text{otherwise}, \end{cases}$$

$$\tilde{Q}_{n+1}^{\gamma}(s, \boldsymbol{l}, \boldsymbol{a}) \tag{2.6}$$
$$= \begin{cases} (1-\alpha_n) \tilde{Q}_n^{\gamma}(s, \boldsymbol{l}, \boldsymbol{a}) + \alpha_n \left[R^{\gamma}(s, \boldsymbol{l}, \boldsymbol{a}) + \beta \cdot \tilde{V}_n^{\gamma}(s_{n+1}) \right], & \text{for } (s, \boldsymbol{l}, \boldsymbol{a}) = (s_n, \boldsymbol{l_n}, \boldsymbol{a_n}), \\ \tilde{Q}_n^{\gamma}(s, \boldsymbol{l}, \boldsymbol{a}), & \text{otherwise}, \end{cases}$$

$$V_{n+1}^{IDS}(s) = \text{NASH}^{IDS}(Q_{n+1}^{IDS}(s, \cdot, \cdot), \tilde{Q}_{n+1}^{\gamma}(s, \cdot, \cdot)), \tag{2.7}$$

$$\tilde{V}_{n+1}^{\gamma}(s) = \text{NASH}^{\gamma}(Q_{n+1}^{IDS}(s, \cdot, \cdot), \tilde{Q}_{n+1}^{\gamma}(s, \cdot, \cdot)). \tag{2.8}$$

The updated strategies of the IDS and attacker are given by

$$\pi_{n+1}^{IDS}(s, \cdot) = \arg \text{NASH}^{IDS}(Q_{n+1}^{IDS}(s, \cdot, \cdot), \tilde{Q}_{n+1}^{\gamma}(s, \cdot, \cdot)), \tag{2.9}$$

$$\tilde{\pi}_{n+1}^{\gamma}(s, \cdot) = \arg \text{NASH}^{\gamma}(Q_{n+1}^{IDS}(s, \cdot, \cdot), \tilde{Q}_{n+1}^{\gamma}(s, \cdot, \cdot)). \tag{2.10}$$

Note that the optimal quality function Q^{IDS} ($\tilde{Q}^{\gamma}$) represents the total expected discounted reward of the IDS (attacker) attained by taking action $\boldsymbol{l}$ ($\boldsymbol{a}$) given the state s and attacker's action $\boldsymbol{a}$ (IDS's action $\boldsymbol{l}$). In (2.7) and (2.9), $\text{NASH}^{IDS}(Q_{n+1}^{IDS}(s, \cdot, \cdot), \tilde{Q}_{n+1}^{\gamma}(s, \cdot, \cdot))$ and $\arg \text{NASH}^{IDS}(Q_{n+1}^{IDS}(s, \cdot, \cdot), \tilde{Q}_{n+1}^{\gamma}(s, \cdot, \cdot))$ give the reward and the corresponding strategy of the IDS at a NE of an equivalent single stage game with corresponding reward functions of the two players specified by the two matrices $Q_{n+1}^{IDS}(s, \cdot, \cdot)$ and $\tilde{Q}_{n+1}^{\gamma}(s, \cdot, \cdot)$, respectively. The reward and the corresponding strategy of the attacker is defined similarly.

It has been shown in [69] that with a suitable learning rate α_n in (2.5) and (2.6), the learned quantities in the Nash-Q algorithm converge to the corresponding optimal ones under certain sufficient conditions. The attacker can learn its optimal strategy through a similar procedure.

2.2.2 Second Layer: Resource Allocation

The second layer is concerned with the resource sharing in the IDN. In this subsection, a VCG auction algorithm is employed, which can be described as follows: First, given the learned strategies from the first layer, each IDS_i reports to the IDN manager the number of libraries it can load currently (i.e., $U_i(s^i)$) and the expected reward function given by

$$\tilde{R}^{IDS_i}(s^i, \boldsymbol{l}^i(U_i), \tilde{\pi}^{\gamma_i}) = \sum_{a_k \in \mathcal{A}} \tilde{\pi}^{\gamma_i}(s^i, a_k) p_{\boldsymbol{l}^i, a_k} w^{IDS_i}_{s^i, a_k}, \tag{2.11}$$

in which $\tilde{\pi}^{\gamma_i}(s^i, a_k)$ denotes the estimated probability that the attacker γ_i will launch attack a_k when the subsystem i is at state s^i. The IDN manager then computes the resource allocation $\boldsymbol{U}' = (U'_1, ..., U'_N)$ that maximizes the total expected reward of all the IDSs by solving the following problem:

$$\begin{aligned} \tilde{R}_{max} = \max_{g_{ij}} \quad & \sum_u \tilde{R}^{IDS_u}(s^u, \boldsymbol{l}^u(U'_u), \tilde{\pi}^{\gamma_u}) \\ \text{s.t.} \quad & \sum_j g_{ij} \leqslant U_i(s^i) \\ & U'_i = U_i(s^i) + \sum_j g_{ji} - \sum_j g_{ij} \\ & g_{ij} \in \mathbb{N}, \ \forall i, j \in \mathcal{N}. \end{aligned} \tag{2.12}$$

Furthermore, the IDN manager computes the maximum total expected reward if IDS_u is excluded from the auction, i.e., $\tilde{R}_{max/u}$ for each $u \in N$. The manager then charges IDS_u by the amount given by

$$\tilde{R}_{max/u} - \sum_{i \neq u} \tilde{R}^{IDS_i}(s^i, \boldsymbol{l}^i(U'_i), \tilde{\pi}^{\gamma_i}). \tag{2.13}$$

Note that for the IDSs who share their resource with the other IDSs, the charges will be negative, which means that the IDN manager will pay them for their effort. The proposed method is summarized in Algorithm 1.

2.2.3 Performance Analysis

In this subsection, some analytical results of the proposed algorithm 1 are presented. Throughout the entire analysis, the following assumptions are made to facilitate the discussion, without loss of generality. To simplify the presentation, it is assumed that the detection probability of library l_m against attack a_k is q_1 when $m = k$ and q_2 otherwise (i.e., $p_{l_m, a_k} = q_1$ if $m = k$, and $p_{l_m, a_k} = q_2$ otherwise), with $q_1 > q_2 > 0$. It is further assumed that the first m IDSs (i.e., $IDS_1, IDS_2, ..., IDS_m$) are in the healthy state and can load $U_1 = U_2 = \cdots = U_m = U(H)$

Algorithm 1 Collaborative Nash-Q Learning for IDS Configuration

1: Initialization: $\{Q_0^{IDS}\} = \mathbf{0}$, $\{\tilde{Q}_0^{\gamma}\} = \mathbf{0}$, $\{V_0^{IDS}\} = \mathbf{0}$, $\{\tilde{V}_0^{\gamma}\} = \mathbf{0}$ and π_0^{IDS}, $\tilde{\pi}_0^{\gamma}$ are uniformly distributed.
2: Each IDS takes action $\boldsymbol{l_n}$ at current state s_n
- uniformly at random with probability p_{explr};
- otherwise, with probability $\{\pi_n^{IDS}(s_n, \boldsymbol{l_n})\}$.

3: Learning: after receiving the reward $\{r_n^{IDS}\}$ and observing the system state transition from s_n to s_{n+1}, each IDS
- update Q^{IDS} and $\tilde{Q}^{\gamma}$ using (2.5) and (2.6), respectively.
- update V^{IDS}, π^{IDS}, $\tilde{V}^{\gamma}$ and $\tilde{\pi}^{\gamma}$ using (2.7), (2.9), and (2.8), (2.10) respectively.

4: Run the resource allocation algorithm.
5: Repeat.

Algorithm 2 Resource Allocation Algorithm: VCG Auction

1: Each IDS senses the current system state $\mathcal{S}_n = \{s_n^1, s_n^2, ..., s_n^N, U_1(s_n^1), U_2(s_n^2), ...U_N(s_n^N)\}$ and obtains the strategies of both its attacker and itself.
2: Each IDS reports to the IDN manager the amount of resource it has and the expected reward function (2.11).
3: After receiving all the reports from IDSs, the IDN manager computes the optimal resource allocation using (2.12).
4: The manager computes the charges for all the IDSs using (2.13).

libraries at a time, while the other IDSs (i.e., $IDS_{m+1}, IDS_{m+2}, ..., IDS_N$) are in the ill state and can only load $U_{m+1} = U_{m+2} = \cdots = U_N = U(I) < U(H)$ libraries simultaneously. Following the same procedure below, the above assumptions may be relaxed and similar results can be obtained for more general cases.

In the following discussion, for the ease of presentation, let $\tilde{\pi}_j^{\gamma_i} = \tilde{\pi}^{\gamma_i}(s^i, a_j)$ denote the estimated probability of attacker γ_i launching attack a_j, and $w_j^{IDS_i} = w_{s^i,a_j}^{IDS_i}$ denote the importance of detecting attack a_j for IDS_i. Define $e_j^i = \tilde{\pi}_j^{\gamma_i} w_j^{IDS_i}$ (here the subscript s^i is omitted because in the resource allocation process, the states of all the subsystems remain unchanged), which admits

$$\sum_{j=1}^{L} \tilde{\pi}_j^{\gamma_i} w_j^{IDS_i} = \sum_{j=1}^{L} e_j^i = k_i, \tag{2.14}$$

where k_i is a constant if $w_j^{IDS_i}$ and $\tilde{\pi}_j^{\gamma_i}$ are known and fixed for $1 \leq j \leq L$. Without loss of

generality, the following ordering $e_1^i \geqslant e_2^i \geqslant ... \geqslant e_L^i$ is assumed, $\forall i \in \mathcal{N}$.[4] Then, it is not difficult to realize that IDS_i is expected to achieve the highest reward if it loads the first U_i libraries, i.e., $l_1, l_2, ..., l_{U_i}$. As a result, the expected reward of IDS_i before resource allocation can be express as

$$E\{R^{IDS_i}|U_i\} = \sum_{j=1}^{U_i} e_j^i[1-(1-q_1)(1-q_2)^{U_i-1}] + \sum_{j=U_i+1}^{L} e_j^i[1-(1-q_2)^{U_i}]. \tag{2.15}$$

Similarly, the expected reward of IDS_i after resource allocation can be expressed as

$$E\{R^{IDS_i}|U_i'\} = \sum_{j=1}^{U_i'} e_j^i[1-(1-q_1)(1-q_2)^{U_i'-1}] + \sum_{j=U_i'+1}^{L} e_j^i[1-(1-q_2)^{U_i'}], \tag{2.16}$$

in which $U_i' = U_i + \sum_j g_{ji} - \sum_j g_{ij}$ denotes the number of libraries IDS_i can load after resource allocation.

Theorem 1. *Let*

$$I = \max_{1\leq t\leq N}\{q_2(1-q_2)k_t - (q_1-q_2)[\sum_{j=1}^{U_t} e_j^t - (1-q_2)\sum_{j=1}^{U_t+1} e_j^t]\}(1-q_2)^{U_t-1}, \tag{2.17}$$

$$D = \min_{1\leq t\leq N}\{q_2k_t - (q_1-q_2)[(1-q_2)^{-1}\sum_{j=1}^{U_t-1} e_j^t - \sum_{j=1}^{U_t} e_j^t]\}(1-q_2)^{U_t-1}. \tag{2.18}$$

Then the resource allocation scheme will lead to improvement in terms of expected reward if and only if the following condition holds:

$$I > D. \tag{2.19}$$

Proof. Please see Appendix A.1. □

Remark 1. *In (2.17) and (2.18), I represents the maximum performance gain of IDN if one of the IDSs receives one unit of resource (one library), while D represents the minimum performance loss of IDN if one of the IDSs gives out one unit of resource. Assume that in a resource allocation scheme which leads to performance improvement, some IDSs give out r units of resource in total, while some other IDSs receive the r units of resource. The concavity of the functions f_j (according to (A.3) in Appendix A.1) indicates that the total performance loss of the IDSs that give out resource is at least rD, while the total performance gain of the other IDSs is at most rI. The necessary condition for performance improvement is thus $rI > rD$, which explains the*

[4] In the more general case, the e^i's can be ordered first and IDS i can load the corresponding libraries after the resource allocation process.

necessity of (2.19). On the other hand, when $I > D$, it means that performance gain can be obtained by simply exchanging one unit of resource, which verifies the sufficiency of (2.19).

The following corollary considers a special but important case.

Corollary 1. *When $q_2 = 0$, i.e., mismatched libraries have no chance to detect attacks successfully, the resource allocation scheme will lead to improvement in terms of expected reward if and only if the following condition holds:*

$$\max_{1\leq t\leq N}\{e^t_{U_t+1}\} > \min_{1\leq t\leq N}\{e^t_{U_t}\}. \tag{2.20}$$

Remark 2. *Note that U_t represents the number of libraries that IDS_t could load before resource allocation. Therefore, the extra expected reward that IDS_t could gain if one more library can be loaded is given by $\sum_{j=1}^{U_t+1} q_1 e^t_j - \sum_{j=1}^{U_t} q_1 e^t_j = q_1 e^t_{U_t+1}$, according to (2.15) and (2.16). Similarly, $q_1 e^t_{U_t}$ represents the loss of expected reward if IDS_t gives out one unit of resource. When the reward gain for a certain IDS is larger than the loss of another, there will be a performance improvement in terms of expected reward by exchanging one unit of resource. In practice, this condition is fairly easy to satisfy, especially when some IDSs face severe attacks (i.e., the probabilities of attacks or the importance of detecting such attacks is high, or equivalently $e^t_{U_t+1}$ is large) while some other IDSs face mild attacks (i.e., a relatively small $e^t_{U_t}$).*

2.3 Distributed Incentive Mechanism Design

Despite the effectiveness of the VCG auction scheme presented above, an IDN manager may not always be available in the case that the collaborative IDSs belong to different institutions. In addition, the centralized architecture assumed by the VCG auction based scheme may suffer from single point of failure, and may not scale well to large systems due to limited computing capability. Considering these, in this section, a distributed game-theoretic incentive mechanism with low computation complexity (as compared to the centralized VCG) is proposed to facilitate effective collaboration among IDSs. Fig. 2.3 depicts the flowchart of the whole resource allocation process. Particularly, the resource allocation process is divided into multiple rounds during which each IDS will allocate all of its resources to all the IDSs (including itself) in the IDN. In each round (indexed by rd), the IDSs with unallocated resource will serve as resource providers, and each resource provider will allocate one unit of resource (and therefore there are $\max(U_i)$ rounds in total) through the proposed incentive mechanism.

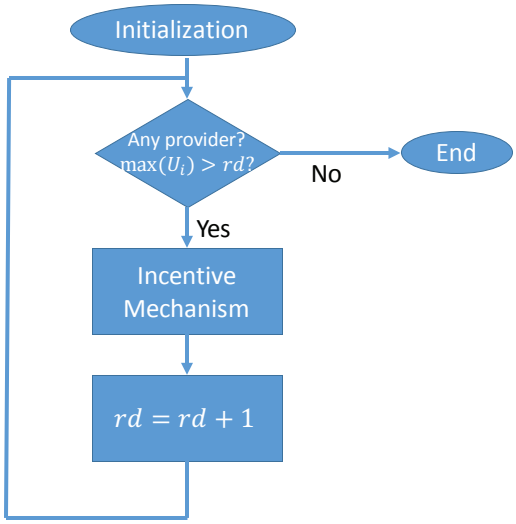

Figure 2.3: Flowchart of the incentive mechanism.

2.3.1 Incentive Model

In this section, the same notations and assumptions are adopted as those of Section 2.2.3. Recall that given the assumption $e_1^i \geqslant e_2^i \geqslant ... \geqslant e_L^i, \forall i \in \mathcal{N}$, when IDS_i obtains k units of resource, its optimal strategy is to load the libraries corresponding to the k-largest e^i's (i.e., $e_1^i, \cdots, e_k^i$).

As mentioned above, the resource allocation process is divided into multiple rounds. Let g_i^j denote the amount of resource that IDS_i allocates to IDS_j.[5] In each round, IDS_i with unallocated resource chooses to allocate one unit of resource to one of the IDSs (i.e., $g_i^j \in \{0,1\}, \sum_{j=1}^{N} g_i^j \leq 1$). However, this leads to an integer programming problem similar to (2.12) which is NP-hard.[6] Therefore, in this section, the condition $g_i^j \in \{0,1\}$ is relaxed to $0 \leq g_i^j \leq 1$. The reward received by IDS_i from allocating resource to IDS_j is given by[7]

$$P_{ji}(\boldsymbol{g^j}) = f_{ji}(\sum_{m=1}^{N} h_{i,m}^j g_m^j) - c_i g_i^j, \tag{2.21}$$

for some pre-specified payment function $f_{ji}(\cdot)$, in which $\boldsymbol{g^j} = [g_1^j, g_2^j, \cdots, g_N^j]$ is a vector denoting the amount of resource that IDS_j has received from each IDS, $h_{i,m}^j$ is the weight of g_m^j that represents the relative importance of the resource received from IDS_m from IDS_i's perspective,

[5]Different from g_{ij} which represents the total amount of resource that IDS_i shares with IDS_j in the VCG auction based scheme, here, g_i^j denotes the amount of resource that IDS_i allocates to IDS_j in each round of the proposed incentive mechanism.

[6]The integer programming problem of VCG auction based collaboration scheme is solved by brute-force search, and therefore has high computation complexity.

[7]If IDS_i has shared all of its resource in the previous rounds, g_i^j is set to $0, \forall j \in \mathcal{N}$.

and $c_i g_i^j$ is the linear cost of IDS_i due to the resource sharing procedure. Intuitively, the payment from IDS_j - the first term in (2.21) - depends on all the resources IDS_j receives in this round, and is expressed in a general form here for flexibility and wider applicability.

For each payment function $f_{ji}(\cdot)$, the following assumptions are made:

Assumption 1. *Each* $f_{ji}(\cdot)$ *is*

- *continuous*
- *strictly increasing*
- *strictly concave on* $[0, \infty]$
- $f_{ji}(0) = 0$
- $f'_{ji}(0) > c_i$.

Some explanations about the practical implications of the above assumptions are in order. The second condition is intuitive since the more resource an IDS receives, the more it will pay to others. The third condition implies diminishing returns for additional resource allocation. The fourth condition suggests that if IDS_j does not receive any resource, it will not pay anything to any other IDSs. The fifth condition requires that the costs of resource sharing (c.f. the second term in (2.21)) should not be too high, otherwise the IDSs have no incentive to collaborate. In this chapter, the following payment functions that conform to these assumptions are considered:

$$f_{ji}(\sum_{m=1}^{N} h_{i,m}^j g_m^j) = [q_1 e_{l_j}^j + q_2 \sum_{l \neq l_j} e_l^j] \ln(\sum_{m=1}^{N} h_{i,m}^j g_m^j + 1), \tag{2.22}$$

in which l_j is the index of library that IDS_j intends to load if it receives one unit of resource in the current round (i.e., IDS_j has obtained $l_j - 1$ units of resource in the previous rounds) and $[q_1 e_{l_j}^j + q_2 \sum_{l \neq l_j} e_l^j]$ is the corresponding expected reward it can obtain by loading library l_j. Note that when the IDSs have comparable e^j's, it makes sense to assume that (almost) all the IDSs will be allocated no more than one unit of resource due to the limited total amount of resource available. When an IDS faces much more severe attacks (or equivalently it sets high importance to attack detection), it is possible that more than one unit of resource can be allocated. In such cases, rigorously speaking, the expected reward should also depend on $e_{l_j+d}^j$ for some $d \geq 1$. However, by the assumption that the e^j's are in a descending order, only the most significant term is considered (i.e., $e_{l_j}^j$) here for tractability. Our simulation results below show that this simplification incurs little performance degradation.

In each round, the objective of IDS_i with unallocated resource is to maximize its total

received reward

$$W_i(\boldsymbol{g_i}) = \sum_{j=1}^{N} P_{ji}(\boldsymbol{g^j}), \tag{2.23}$$

in which $\boldsymbol{g_i} = [g_i^1, g_i^2, \cdots, g_i^N]$ is a vector that denotes the amount of resource that IDS_i allocates to each IDS (at most one unit in each round). To find the optimal resource sharing stretegy, each IDS needs to solve the following (concave) optimization problem:

$$\begin{aligned} \max_{g_i^j, \forall j \in \mathcal{N}} \quad & \sum_{j=1}^{N} \Big[[q_1 e_{l_j}^j + q_2 \sum_{l \neq l_j} e_l^j] \ln(\sum_{m=1}^{N} h_{i,m}^j g_m^j + 1) - c_i g_i^j \Big] \\ \text{s.t.} \quad & \sum_{j} g_i^j \leqslant 1 \\ & g_i^j \geq 0, \ \forall j \in \mathcal{N}. \end{aligned} \tag{2.24}$$

Note that the decision variable of each IDS_j is a vector $\boldsymbol{g_i}$ and the action sets of IDSs are not coupled. Therefore, Lagrangian relaxation is used to penalize the constraints to solve the optimization problem. The Lagrangian $L_i(\boldsymbol{g_i}, \lambda_i)$ of IDS_i's optimization problem is given by

$$L_i(\boldsymbol{g_i}, \lambda_i) = \sum_{j=1}^{N} [q_1 e_{l_j}^j + q_2 \sum_{l \neq l_j} e_l^j] \ln(\sum_{m=1}^{N} h_{i,m}^j g_m^j + 1) - c_i \sum_{j=1}^{N} g_i^j - \lambda_i (\sum_{j=1}^{N} g_i^j - 1), \tag{2.25}$$

in which λ_i is the Lagrange multiplier. By Lagrangian relaxation, the action set is reduced to $G_i = \{\boldsymbol{g_i} | g_i^j \geq 0, \forall j \in \mathcal{N}\}$. The above Lagrangian will be considered as the reward function of the relaxed resource allocation game, and the corresponding NE properties will be discussed in the next subsection.

2.3.2 Equilibrium Properties

As a common approach in the literature (e.g., [5, 70]), pure strategy Nash Equilibrium is considered in this chapter.

Definition 1. *[71] For a given Lagrangian multiplier λ_i, a Nash equilibrium (NE) $\{\boldsymbol{g_i^*}, i \in \mathcal{N}\}$ for the relaxed game is a set of strategies that satisfy*

$$L_i(\boldsymbol{g_i^*}, \boldsymbol{g_{-i}^*}, \lambda_i) \geq L_i(\boldsymbol{g_i}, \boldsymbol{g_{-i}^*}, \lambda_i), \forall g_i \in A_i, i \in \mathcal{N}, \tag{2.26}$$

in which $\boldsymbol{g_{-i}} = \{\boldsymbol{g_v} : v \neq i, v \in \mathcal{N}\}$ is comprised of the resource sharing decision vectors of other IDSs.

Optimality Conditions

To solve the relaxed optimization problems in (2.25) for the IDSs, the equivalence between finding the NE and solving the corresponding Linear Complementarity Problems (LCPs) [72] is illustrated first.

Specifically, given the relaxed optimization problems and their corresponding Lagrangians, applying the first-order KKT condition to each IDS's optimization problem yields

$$\frac{\partial L_i}{\partial g_i^j} = 0, \forall i, j \in \mathcal{N}, \tag{2.27}$$

which implies that

$$\sum_{m=1}^{N} h_{i,m}^j g_m^j = \frac{[q_1 e_{l_j}^j + q_2 \sum_{l \neq l_j} e_l^j]}{c_i + \lambda_i} - 1, \forall i, j \in \mathcal{N}. \tag{2.28}$$

It also follows from KKT that for any equilibrium point, g_i^j must satisfy

$$\begin{aligned} \sum_{m=1}^{N} h_{i,m}^j g_m^j &= \frac{[q_1 e_{l_j}^j + q_2 \sum_{l \neq l_j} e_l^j]}{c_i + \lambda_i} - 1 \quad if\ g_i^j > 0, \\ \sum_{m=1}^{N} h_{i,m}^j g_m^j &\geq \frac{[q_1 e_{l_j}^j + q_2 \sum_{l \neq l_j} e_l^j]}{c_i + \lambda_i} - 1 \quad if\ g_i^j = 0, \end{aligned} \tag{2.29}$$

and by concavity of f_{ji} made previously, these conditions are also sufficient. To simplify the notations, define a set of weight matrices $\boldsymbol{H^{(j)}}$'s and vectors $\boldsymbol{b^{(j)}}$'s as follows:

$$\begin{aligned} H_{mn}^{(j)} &= [h_{m,n}^j], \quad \forall m, n \in \mathcal{N}, \\ b_i^{(j)} &= \frac{[q_1 e_{l_j}^j + q_2 \sum_{l \neq l_j} e_l^j]}{c_i + \lambda_i} - 1, \quad \forall i, j \in \mathcal{N}. \end{aligned} \tag{2.30}$$

Then the optimality conditions can be rewritten as the following LCP

$$\begin{aligned} &\boldsymbol{y^j} = \boldsymbol{H^{(j)}} \boldsymbol{g^j} - \boldsymbol{b^{(j)}} \\ &\boldsymbol{y^j}^T \boldsymbol{g^j} = \boldsymbol{0} \\ &\boldsymbol{g^j} \geq \boldsymbol{0}, \boldsymbol{y^j} \geq \boldsymbol{0}, \forall j \in \mathcal{N}. \end{aligned} \tag{2.31}$$

Since the above conditions are both necessary and sufficient, it follows that finding a NE for the game is equivalent to solving the corresponding LCPs.

Existence and Uniqueness of NE

The existence of NE guarantees that by following the proposed distributed incentive mechanism, each IDS can finally obtain a resource allocation strategy that maximizes its own reward given that other IDSs are rational. The uniqueness of NE guarantees that such resource allocation strategy is unique, and hence there will be no dispute among the selfish IDSs about equilibrium strategy selection.

Definition 2. *A matrix* $\boldsymbol{W} \in \mathbb{R}^{n\times n}$ *is strictly diagonally dominant if* $\sum_{m\neq n} |W_{nm}| < |W_{nn}| = 1, \forall n$.

Definition 3. *A complex square matrix* $\boldsymbol{W} \in \mathbb{R}^{n\times n}$ *is a P-matrix if its every principal minor is strictly positive.*

In order to motivate the IDSs to collaborate and share their resource with each other, in the proposed distributed incentive mechanism, $\boldsymbol{H}^{(j)}$'s are chosen to be diagonally dominant for all $j \in \mathcal{N}$. In this case, $h^j_{i,i} > h^j_{i,m}, \forall i, j, m \neq i \in \mathcal{N}$. It can be verified (c.f. the best response of each IDS given by (2.29)) that such a setting can promote resource sharing among collaborative IDSs. Particularly, the matrices $\boldsymbol{H}^{(j)}$'s are set to

$$\boldsymbol{H}^{(j)}{}_{mn} = \begin{cases} 1 & \text{if } m = n, \\ \frac{1}{N} & \text{if } m \neq n. \end{cases}$$

For the matrices $\boldsymbol{H}^{(j)}$'s considered above, the following theorem holds:

Theorem 2. *The given game admits a unique Nash Equilibrium.*

Proof. Please see Appendix A.2. □

2.3.3 Iterative Algorithm

Intuitively, when the collaboration mechanism is at the NE, no IDS will deviate. When it is not at the NE, then each IDS may choose to update its sharing strategies based on the current strategies of other IDSs. Formalizing this intuition, a dynamic algorithm to compute the unique NE is introduced in this subsection to compute the NE.

Let $g^j_i(t)$ denote the amount of resource that IDS_i allocates to IDS_j at time step t. Consider the following asynchronous update rule:

$$g^j_i(t+1) = \begin{cases} \max(0, \frac{[q_1 e^j_{l_j} + q_2 \sum_{l\neq l_j} e^j_l]}{c_i+\lambda_i} - 1 - \frac{1}{N}\sum_{n\neq i} g^j_n) & \text{if } t+1 \in T^i, \\ g^j_i(t) & \text{if } t+1 \notin T^i, \end{cases}$$

in which T^i is the set of times at which IDS_i updates its sharing strategy. Suppose that all the IDSs follow the asynchronous update rule and broadcast their sharing strategies using their own schedule, and assume that the time sets T^i's are infinite for the IDSs, which means all the IDSs will update infinitely often. The following theorem holds:

Theorem 3. *Suppose that $\boldsymbol{H^{(j)}}$'s are diagonally dominant for all $j \in \mathcal{N}$. Then the asynchronous update algorithm converges to the unique game NE from any starting point $g_i^j > 0, \forall i, j \in \mathcal{N}$.*

The convergence of the asynchronous update algorithm can be proved using similar a method as in [73]. Moreover, in the proposed incentive mechanism, the algorithm described above depends on the Lagrange multiplier λ_i. Therefore, duality can be exploited to devise an iterative algorithm for the Lagrange multiplier [5]. Let $D_i(\lambda_i)$ be the dual function given by

$$D_i(\lambda_i) = \max_{\boldsymbol{g_i}} L_i(\boldsymbol{g_i}, \lambda_i). \tag{2.32}$$

According to (2.25), the dual function is given by

$$D_i(\lambda_i) = \sum_{j=1}^{N} [q_1 e_{l_j}^j + q_2 \sum_{l \neq l_j} e_l^j] \ln(\boldsymbol{H}^{(j)} \boldsymbol{g}^j + 1) - c_i \sum_{j=1}^{N} g_i^j - \lambda_i (\sum_{j=1}^{N} g_i^j - 1). \tag{2.33}$$

It is easy to realize that $D_i(\lambda_i)$ is a convex function and a dual optimal λ_i^* solves the dual optimization problem given by

$$\min_{\lambda_i > 0} D_i(\lambda_i). \tag{2.34}$$

By using (2.29) and taking the first-order derivative of the dual function, it is easy to find that

$$D_i'(\lambda_i) = 1 + \sum_{j \in C_i} [1 + \frac{1}{N} \sum_{n \neq i} g_n^j - \frac{[q_1 e_{l_j}^j + q_2 \sum_{l \neq l_j} e_l^j]}{c_i + \lambda_i}], \tag{2.35}$$

in which $C_i = \{j \in \mathcal{N} | g_i^j > 0\}$.

By further taking the second-order derivative of the dual function, it can be obtained that

$$D_i''(\lambda_i) = \sum_{j \in C_i} \frac{[q_1 e_{l_j}^j + q_2 \sum_{l \neq l_j} e_l^j]}{(c_i + \lambda_i)^2}. \tag{2.36}$$

Note that given (2.36), $D_i^{''}(\lambda_i) \geq 0$ always holds, with $D_i^{''}(\lambda_i) = 0$ if and only if $g_i^j = 0, \forall j \in \mathcal{N}$. Furthermore, for any resource provider IDS_i, the cost c_i is upper-bounded by our assumption. Therefore, it can be concluded that the dual function is strongly convex, i.e., the corresponding Hessian is bounded $B_1 \leq D_i^{''}(\lambda_i)$, for some strictly positive B_1 [74].

Due to convexity, the dual optimal λ_i^* can be computed by applying $D_i'(\lambda_i) = 0$, which is

obtained as follows:

$$\lambda_i^* = \frac{\sum_{j \in C_i}[q_1 e_{l_j}^j + q_2 \sum_{l \neq l_j} e_l^j]}{1 + \sum_{j \in C_i}[1 + \frac{1}{N} \sum_{n \neq i} g_n^j]} - c_i. \tag{2.37}$$

To achieve the dual optimal, the following gradient search can be used to find the optimal λ_i^*:

$$\lambda_i(t+1) = \lambda_i(t) - \beta_i D_i'(\lambda_i(t)), \forall i \in \mathcal{N}, \tag{2.38}$$

in which $\beta_i \in (0, 1)$ is the step size.

Theorem 4. *$D_i'(\lambda_i)$ is Lipschitz continuous with some strictly positive Lipschitz constant K, and the gradient based iterative algorithm converges to dual optimal λ_i^* with step size $\beta_i < \frac{min(2, B_1)}{K}$.*

Proof. Please see Appendix A.3. □

Based on the discussion above, the asynchronous update algorithm for the proposed incentive mechanism is summarized in Algorithm 3. During the asynchronous updating process, in addition to the amount of resource U_i and the reward function given by (2.21), each IDS_i also broadcasts the amount of resource g_i^j allocated to another IDS_j as well as the Lagrange multiplier λ_i.

Algorithm 3 Resource Allocation Algorithm: Distributed Incentive Mechanism

1: Initialization: obtain U_i and e_k^i for $1 \leq i \leq N, 1 \leq k \leq L$, from the first layer game and IDS_i broadcasts the obtained information, $\forall i$
2: Set $rd = 0$
3: **while** $rd < max(U_i)$ **do**
4: Determine the resource providers set $\mathcal{P}$ $(U_i > rd)$
5: Run the asynchronous update algorithm
6: Run the heuristic resource allocation algorithm
7: $rd = rd + 1$
8: **end while**

Note that the resource allocation strategies given by the distributed incentive mechanism are continuous. However, in the final resource allocation scheme, the amount of resource that each IDS receives should be integers. To mitigate this problem, a heuristic resource allocation algorithm is proposed as given by Algorithm 5. In each round, the total amount of resource that each IDS has received is computed and compared. Then one unit of resource is allocated to the IDS which receives the most resource according to Algorithm 3, followed by a procedure which updates the total amount of received resource.

Algorithm 4 Asynchronous Update Algorithm

Initialization: set $t = 0$, $g_i^j = 0$, $\lambda_i = 1$, $\forall i \in \mathcal{P}, j \in \mathcal{N}$
repeat
  for $i = 1, \cdots, N$ **do**
    if $t \in T^i$ and $i \in \mathcal{P}$ **then**
      $g_i^j(t+1) = \max(0, \frac{[q_1 e_{l_j}^j + q_2 \sum_{l \neq l_j} e_l^j]}{c_i + \lambda_i} - 1 - \frac{1}{N} \sum_{n \neq i} g_n^j)$
      $\lambda_i(t+1) = \lambda_i(t) - \beta_i D_i'(\lambda_i(t))$
    else
      $g_i^j(t+1) = g_i^j(t)$
      $\lambda_i(t+1) = \lambda_i(t)$
    end if
  end for
  $t = t + 1$
until converged

Algorithm 5 Heuristic Resource Allocation Algorithm

Given $g_i^j, \forall i \in \mathcal{P}, j \in \mathcal{N}$, let $\tilde{g}^j = \sum_i g_i^j$
for $n = 1, \cdots, |\mathcal{P}|$ **do**
  $k = \text{argmax}_m[\tilde{g}^m]$
  $U_k' = U_k' + 1$
  $\tilde{g}^k = \tilde{g}^k - \frac{\sum_j \tilde{g}^j}{|\mathcal{P}|}$
end for

2.4 Simulation Results

This section presents the simulation results to evaluate and compare the effectiveness of the proposed algorithms in different scenarios. Specifically, in all the scenarios, unless otherwise noted, it is assumed that there are three IDSs, with three corresponding attackers. It is further assumed that the detection libraries available to all the IDSs are $\mathcal{L} = \{l_1, l_2, ..., l_8\}$, and the attacking libraries available to all the attackers are $\mathcal{A} = \{a_1, a_2, ..., a_8\}$. The detection probability p_{l_i,a_k} of library l_i against attack a_k is 0.85 (q_1) when $i = k$ and 0.1 (q_2) otherwise. It is further assumed that an IDS can load 1 and 5 libraries when it is at the ill and the healthy states, respectively, i.e., $U(I) = 1$, $U(H) = 5$. The discounting factor β, exploration probability p_{explr} and the learning rate for the Nash-Q learning algorithms of both IDSs and attackers are chosen according to [69, 75]. The average accumulated reward $\bar{r}_n^{IDS}$ defined below is considered as the performance metric of interest:

$$\bar{r}_n^{IDS} = \frac{1}{n} \sum_{i=1}^{n} r_i^{IDS}. \tag{2.39}$$

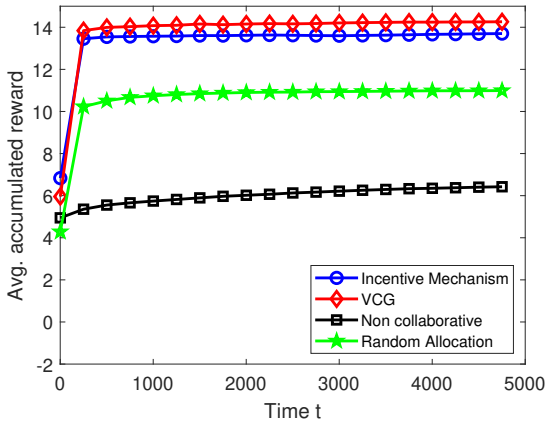

Figure 2.4: The overall average accumulated reward of IDSs.

2.4.1 Performance of the proposed algorithms

First, the performance of the proposed algorithms in terms of the overall average accumulated reward of the three IDSs in an exemplary case is examined and compared. It is assumed that IDS_1 and IDS_2 face attackers that can only launch one attack while IDS_3 faces an attacker that can launch five attacks simultaneously. The importance factors of detecting different attacks for IDS_1 and IDS_3 are set as $w^{IDS_1} = \left[\begin{smallmatrix} 1 & 1 & 2 & 3 & 1 & 1 & 3 & 15 \\ 4 & 2 & 1 & 2 & 3 & 1 & 1 & 9 \end{smallmatrix}\right]$ and $w^{IDS_3} = \left[\begin{smallmatrix} 11 & 12 & 11 & 15 & 17 & 2 & 1 & 1 \\ 12 & 14 & 23 & 24 & 17 & 1 & 3 & 2 \end{smallmatrix}\right]$, with the number at the i-th row and the j-th column representing the importance of detecting attack a_j at state i (i=1 for healthy and i=2 for ill). Similarly, the profit factors for attackers are set as $w^{\gamma_1} = \left[\begin{smallmatrix} 1 & 1 & 1 & 3 & 1 & 1 & 2 & 10 \\ 3 & 2 & 1 & 4 & 3 & 1 & 8 & 13 \end{smallmatrix}\right]$ and $w^{\gamma_3} = \left[\begin{smallmatrix} 11 & 13 & 12 & 12 & 15 & 5 & 3 & 1 \\ 12 & 17 & 13 & 14 & 16 & 2 & 4 & 1 \end{smallmatrix}\right]$. Besides, for IDS_1, the corresponding action-dependent state transition matrices are set as $p_{st}(\cdot|\cdot) = \left[\begin{smallmatrix} 1 & 0 \\ 0.5 & 0.5 \end{smallmatrix}\right]$ if the corresponding attack is detected and $p_{st}(\cdot|\cdot) = \left[\begin{smallmatrix} 0.5 & 0.5 \\ 0.2 & 0.8 \end{smallmatrix}\right]$ otherwise, with the number at the i-th row and the j-th column representing the probability of transiting from state i to state j. For IDS_3, the corresponding action-dependent state transition matrices are set as $p_{st}(\cdot|\cdot) = \left[\begin{smallmatrix} 1 & 0 \\ 0.5 & 0.5 \end{smallmatrix}\right]$ if all the 5 attacks are detected, $p_{st}(\cdot|\cdot) = \left[\begin{smallmatrix} 0.5 & 0.5 \\ 0.2 & 0.8 \end{smallmatrix}\right]$ if 4 attacks are detected, $p_{st}(\cdot|\cdot) = \left[\begin{smallmatrix} 0.2 & 0.8 \\ 0.1 & 0.9 \end{smallmatrix}\right]$ if 3 attacks are detected, $p_{st}(\cdot|\cdot) = \left[\begin{smallmatrix} 0.1 & 0.9 \\ 0.05 & 0.95 \end{smallmatrix}\right]$ if 2 attacks are detected, $p_{st}(\cdot|\cdot) = \left[\begin{smallmatrix} 0.05 & 0.95 \\ 0.01 & 0.99 \end{smallmatrix}\right]$ if only 1 attack is detected, and $p_{st}(\cdot|\cdot) = \left[\begin{smallmatrix} 0.01 & 0.99 \\ 0 & 1 \end{smallmatrix}\right]$ if no attack is detected. The parameters of IDS_2 are set to be the same as IDS_1; The overall average accumulated rewards of using the proposed algorithms are compared with the original Nash-Q Learning Algorithm without the resource allocation step and a collaboration scheme in which the IDSs randomly allocate their redundant resources. It can be observed from Fig. 2.4 that the performance of both of the proposed algorithms (denoted by "Incentive Mechanism" and "VCG", respectively) significantly outperforms the counterparts "Non-collaborative" and "Random Allocation", with performance improvement of around 100% and 25%. Similar trends are observed when different sets of system

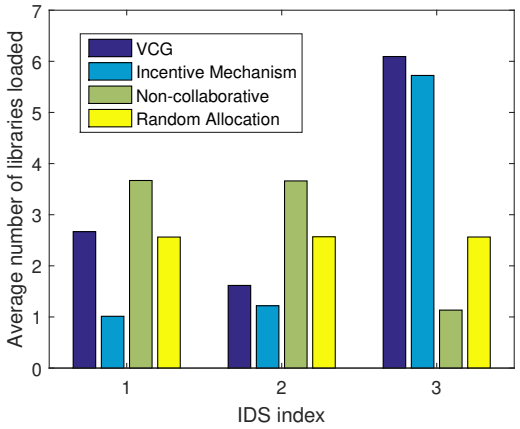

Figure 2.5: Average number of libraries loaded by each IDS in the three IDSs case.

parameters are used (the results are omitted in the interest of space). As expected, the more spare resource the IDSs have, the larger the collaboration gain is. As can be seen from Fig. 2.5, comparing to the non-collaborative case, IDS_3 who faces the most severe attack cam load around 5 more libraries on average. While IDS_1 and IDS_2 sacrifice some performance (since their average number of libraries loaded decreases compared to the "Non-collaborative case"), the overall performance is significantly improved due to the dramatic improvement at the IDN bottleneck, IDS_3 in this case (because IDS_3 faces a more aggressive attacker and therefore is more likely to be compromised). In addition, it can be seen from both Fig. 2.4 and Fig. 2.5 that the performance of the proposed distributed incentive mechanism is close to the socially optimal outcome given by the VCG based scheme.

2.4.2 Improvement Against Aggressiveness of the Attacker

In this section, we examine the impact of the attacker's aggressiveness on the proposed algorithms. It is assumed that IDS_1 and IDS_2 face attackers who can only launch one attack, but IDS_3 faces a more aggressive attacker who can launch multiple attacks. It is shown in Table 2.2 that when IDS_3 faces a more aggressive attacker, the collaboration gains of both proposed algorithms in terms of improvement in the overall average accumulated reward become more significant. Intuitively, when one of the IDSs faces more attacks, it is more important for this IDS to acquire help from others. Moreover, the performance differences between the proposed incentive mechanism and VCG auction based collaboration scheme is only around 7%.

Table 2.2: The improvement in reward against aggressiveness of attackers

Number of attacks on IDS_3	3	4	5	6
Improvement-Incentive mechanism	70%	80%	100%	113%
Improvement-VCG	77%	86%	107%	120%

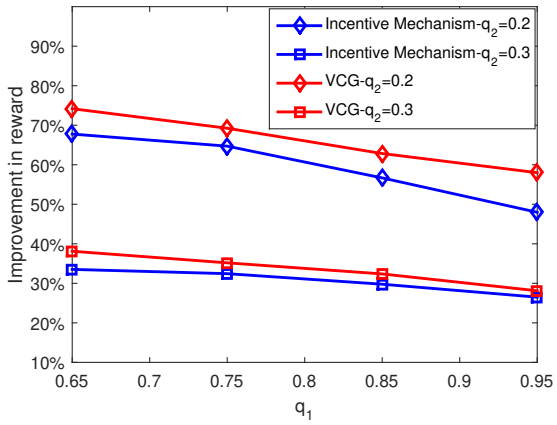

Figure 2.6: The improvement in reward against q_1 and q_2.

2.4.3 Improvement with Respect to Detection Probabilities

In this section, the impact of detection probabilities of libraries against attacks (i.e., q_1 and q_2) are examined.[8] Again, assume that IDS_1 and IDS_2 face attackers who can only launch one attack at a time, and IDS_3 faces an attacker who can launch 5 attacks simultaneously. Fig. 2.6 shows that the performance of both proposed algorithms degrades as q_2 increases. This may be explained as follows: when q_2 is larger, the probability of detecting attacks with mismatched libraries increases, thus reducing the need for collaboration among IDSs. It can also be observed that in all the examined cases, the performance gap between the proposed distributed incentive mechanism and the VCG auction based collaboration scheme is no larger than 10%, which indicates the effectiveness of the incentive mechanism.

In addition, Table 2.3 shows the probability of the condition in Corollary 1 being satisfied when $q_2 = 0$, in which the probability is the number of time slots that the condition is satisfied over the total number of time slots. It can be observed that it is almost always satisfied for all the simulated scenarios, which indicates the effectiveness of the proposed VCG action based collaboration scheme.

[8]Note that the learned strategies (i.e, $\tilde{\pi}^{\gamma}$ in (2.14)) depend on q_1 and q_2. The results in this section show the impact of q_1 and q_2 on the conditions under which the proposed algorithm leads to improvement.

Table 2.3: The probability of the condition in Corollary 1 being satisfied

q_1	0.65	0.75	0.85	0.95
Probability	99.82%	99.65%	99.57%	99.49%

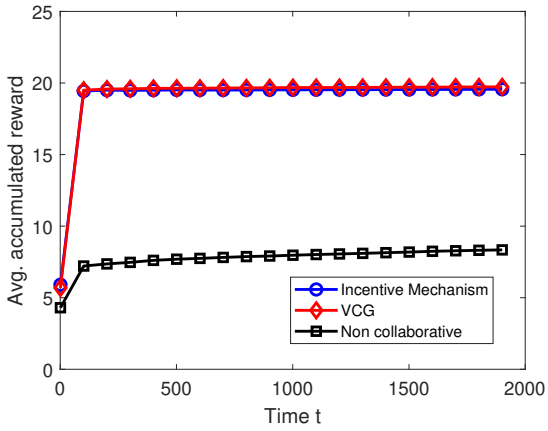

Figure 2.7: The overall average accumulated reward of IDSs.

2.4.4 Performance of the proposed algorithms with more IDSs

In this section, an IDN with ten IDSs is considered. The parameters of the first five IDSs are set to be the same as those of IDS_1 in Section 2.4.1, while the parameters of the last five IDSs are set to be the same as those of IDS_3. In addition, the first five IDSs face attackers that can only launch one attack, and the last five IDSs face attackers that can launch five attacks simultaneously. Similar to the three IDSs case, it can be observed from Fig. 2.7 that both of the proposed algorithms (denoted by "Incentive Mechanism" and "VCG", respectively) significantly outperform the "Non-collaborative" baseline, and the relative performance gains are about 120%. In this case, it can be observed that the performance of the proposed distributed incentive mechanism is closer to the socially optimal performance given by the VCG based scheme, which further verifies its effectiveness.

2.5 Related Works

In the past decades, various game-theoretic approaches have been applied to predict the behavior of intruders and thus improve the detection performance of IDSs. Early works in this direction often model the interaction between a single IDS and an intruder as a two-player game [76–78]. These studies mainly focus on determining the optimal strategies of the IDS and predicting the

possible actions of the intruder by analyzing the NE of the corresponding two-player game. For example, in [76], a two-player, nonzero-sum, non-cooperative game is formulated to facilitate the IDS to form an optimal defense strategy for the sensor network, based on the corresponding NE. [77] studies a network environment where multiple intrusion detection techniques are deployed, and the NE was used to guide the IDS to choose among the different intrusion detection techniques. In addition, considering that the target systems are often dynamic in practice, [78] formulates a multi-stage dynamic intrusion detection game in which the IDS maintains and updates beliefs about the intruders by using Bayesian rules, and the resulting Perfect Bayesian equilibrium (PBE) specifies the best response strategies of the IDS in each stage. Besides, SG has been employed in [64] to study the IDS configuration problem. Our previous work [75] further considers incomplete information due to uncertainty in the intruder's type.

In this chapter, the same model as the ones in [64] and [75] is used to model the configuration problem. This model is rather general and can be used to model a wide range of signature-based IDSs, among which a popular one is Snort. Snort performs rule-based content pattern matching and can detect a variety of attacks and probes, such as buffer overflows, stealthy port scans, and much more [79]. Upon scrutinizing a packet, Snort compares the features (e.g., IP address, payloads) of the packet with the rules defined in the configuration file and generates alerts if necessary. The rules can be downloaded online or designed by a host itself. Essentially, different rules have different detection performance against the same attack [80]. In addition, a host-based IDS usually has only limited resources (e.g., CPU execution time) available from the host system. Even worse, if the host is under attack (e.g., resource exhaustion attacks), even fewer computational resources can be allocated to the IDS. In this case, if too many rules are considered, packet drop will occur because Snort does not have the capacity to examine the packets against all the rules fast enough. In our model, a library l is a set of rules for a specific attack; the subsystem state s measures the security state of the host. For example, the ill state indicates that the IDS failed to detect some previously launched attacks (e.g., CPU exhaustion attacks) and therefore fewer libraries can be loaded. The goal of a host is to find the optimal configuration of rules under resource constraints against a smart attacker. In this sense, [64] can be considered as the non-collaborative case of our model, which serves as a baseline in the simulation.

In practice, multiple IDSs may be deployed in a network and therefore these IDSs can collaborate to improve the overall detection performance. Some collaborative intrusion detection systems (CIDSs) have been proposed in the literature (see [3] and the references therein). Most of the existing CIDSs consider the scenario in which multiple local monitors share their network traffic or alert information with either the collaborative peers (in the distributed case) or a central node (in the centralized case), which will further aggregate the shared information

for more accurate detection. In particular, the IDSs in these CIDSs collaborate to detect the same attack. However, in practice, different hosts (e.g., companies and organizations) may face different attackers. With such consideration, another form of collaboration is considered in this chapter, in which the IDSs (and equivalently the hosts) share their computational resources to help detect the intrusions for the IDSs with insufficient resources. The consultation-base collaborative model in [5, 6, 81, 82] is adapted to model the collaboration among the IDSs in this chapter. In particular, when an IDS predicts severe potential attacks, it can send requests to the other IDSs such that those more powerful IDSs can help them by allocating their redundant resources to do the intrusion detection and responding to the requests. [5, 6, 81, 82] mainly focus on trust management models and incentive designs to overcome the system vulnerability to malicious peers in CIDSs and therefore provide more effective collaboration. However, none of these works consider that the performance of an IDS may be severely degraded when it is in an ill state and does not have sufficient resources to perform intrusion detection. In addition, none of the existing works considers both the configuration problem and the collaboration problem simultaneously. In this chapter, a Nash-Q learning based algorithm is proposed, combined with a distributed incentive mechanism which approaches the socially optimal outcome given by the VCG auction in performance.

2.6 Conclusions

In this chapter, the collaborative IDS configuration problem against rational attackers is tackled through a two-layer SG approach. To solve the two-layer SG, the VCG auction based collaboration scheme is proposed. To further mitigate the communication overhead and complexity issues in VCG auction, a distributed game-theoretic incentive mechanism is also proposed. Analytical and simulation results show that the proposed algorithms can both provide effective collaborative configurations and deliver substantial performance gain as compared to the non-collaborative counterpart.

Chapter 3

On the Security-Privacy Tradeoff in Collaborative Security: A Quantitative Information Flow Game Perspective

While improving the overall security performance, the security-related information exchange leads to potential privacy leakage. As a result, it is essential for collaborative entities to better understand the potential privacy leakage risk in the collaboration. In this chapter, the security-privacy tradeoff in collaborative security is investigated, in which the Quantitative Information Flow (QIF) theory is adopted to quantitatively measure the amount of preserved privacy and the corresponding utility loss. Three QIF games are formulated, each corresponding to one possible scenario of interest in practice. In particular, we first consider the scenario that the entities in the network are fully collaborative, which means that they act as a whole and take the privacy of the whole network into consideration. In this case, the interaction between the collaborative entities and the attacker is modeled as a zero-sum game. Then, considering that the entities and the attacker may give different weights to learning the other's secret, the zero-sum game is further generalized into a non-zero-sum one. Finally, considering that the entities may be selfish and concerned only about their own privacy, we formulate the interaction between the entities and the attacker as a two-layer non-fully collaborative game. The first-layer leader-follower game models the interaction between the attacker and the entities, in which the attacker acts as the leader and the entities act as the followers. The second-layer game models the interaction among the collaborative entities. By solving the QIF games, the optimal attacking strategy for the attacker and the optimal collaboration strategies for the entities in different settings are

Table 3.1: Important notations

$\mathcal{N}$	the set of collaborative entities.
s	state of the network.
$a_1(a_2)$	the attacker launches (does not launch) an attack
π^A	attacking strategy of attacker.
$\mathcal{Q}^j$	observation capability of entity j.
Y_j	observation of entity j.
p_j^c	misreport probability of entity j.
$\hat{Y}_j$	obfuscated (shared) observation of entity j.
$\boldsymbol{Y}$	the true observations of all the entities.
$\hat{\boldsymbol{Y}}$	the obfuscated observations received by the leader.
V	prior vulnerability.
$\hat{V}_X$ ($\hat{V}_{Y_j}$)	posterior vulnerability of the attacker (entity j).

obtained.

The remainder of this chapter is organized as follows. Section 3.1 formulates the collaborative security problem. Some preliminary knowledge about QIF is introduced in Section 3.2. The proposed games are formulated and solved in Section 3.3. A case of study is provided in Section 3.4. Related works are discussed in Section 3.5, and Section 3.6 concludes this chapter.

3.1 Problem Formulation

In this chapter, a network that consists of N different security entities is considered, denoted by $\mathcal{N} = \{1, 2, ..., N\}$. Let s denote the state of the network. Important notations used in this chapter are summarized in Table 3.1.

3.1.1 Attacker Model

An external attacker that can infer and adapt to the collaborative strategies of the security entities is considered. It is assumed that the attacker is able to manipulate the state of the network by launching attacks and its goal is to attack the network without being detected. For the ease of presentation, the following discussion will be focused on one type of attack (e.g., Phishing attack).[1] As a result, there are two possible network states, i.e., $s \in \{0, 1\}$ in which $s = 1$ ($s = 0$) stands for the abnormal (normal) state, corresponding to the case that the attacker launches (does not launch) an attack. To this end, the attacker has to determine a probability distribution $\boldsymbol{\pi^A} = [\pi^A(a_1), \pi^A(a_2)]$, in which $\pi^A(a_1)$ ($\pi^A(a_2) = 1 - \pi^A(a_1)$) is the probability

[1]The model can also be easily extended to the setting of multiple attack types by enlarging the network state space.

that the attacker chooses to launch (not to launch) an attack. In this sense, the action space of the attacker is given by $\mathcal{A} = \{\boldsymbol{\pi}^{\boldsymbol{A}} | 0 \leq \pi^A(a_1), \pi^A(a_2) \leq 1\}$. The cost and reward of the attacker will be discussed in Section 3.3. In addition, it is assumed that the collaborative entities face a powerful attacker that can obtain (e.g., by eavesdropping [83]) the information shared among the entities in the network (i.e., the obfuscated observations and the misreport probabilities).

3.1.2 Defender Model

Each entity (i.e., defender) j in the network independently obtains a private observation (denoted by Y_j) about the network state s and the goal of the entities is to estimate the true network state in a collaborative manner. Each entity j knows the probability distribution of its private observation, which is represented by a set of parameterized marginal distributions $\mathcal{Q}^j = \{q_j(Y_j|s) | Y_j \in \{0, 1\}\}$, where $q_j(\cdot|s)$ is the distribution of the private observation given the true network state s. For example, in collaborative intrusion detection, $q_j(Y_j = 1|s = 1)$, $q_j(Y_j = 0|s = 1)$ and $q_j(Y_j = 1|s = 0)$ correspond to the detection rate, false negative rate and false positive rate of IDS j, respectively. Note that $\mathcal{Q}$'s (i.e., $\{\mathcal{Q}^j\}_{j=1}^N$) are typically known to the collaborators (e.g., the detection rate of an IDS can be captured by its ROC curve, which can be shared among the entities in the network) while the attacker can also obtain such information from past experience. With such consideration, the $\mathcal{Q}$'s are considered common knowledge to both the collaborative entities and the attacker.

Since the private observation may not be sufficient for each entity to learn the true network state s individually, this chapter considers the scenario in which the entities in the network can collaborate and share their observations to further enhance the network security. However, considering that the observations are private, such observation sharing will lead to potential privacy leakage for the entities, particularly under the assumption that the shared information may be eavesdropped by the attacker. Specifically, by revealing the exact observations, additional security-related information of an entity (e.g., the fact that the attacker has already been detected and therefore some critical parts of an entity's system may be monitored) may be inferred by the attacker. Given the information that some critical parts of an entity's system are monitored, the attacker can design better future attacking strategy targeting this entity [84]. Given the true detection result, the attacker that launches the attack can further explore whether it has already been detected and determine its follow-up moves (e.g., leave the system to avoid being identified and backtracked, or discard the data obtained during the attack which has possibly been altered by the entity). Note that if an entity detects the attack successfully, instead of removing the attacker immediately, the entity can pretend that the attack succeeds and provide false data to mislead the attacker and collect the attacker's information (e.g., IP address), which is called deception in the literature [85]. The leakage of the detection results renders such deception

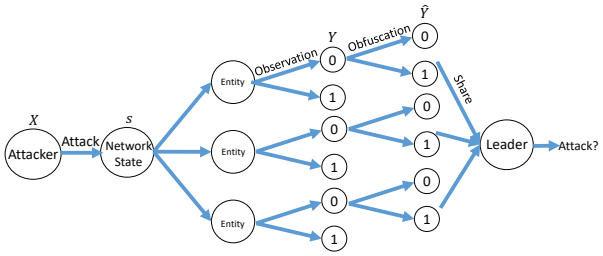

Figure 3.1: Block diagram of the system model.

techniques useless.[2]

In order to preserve privacy, each entity j shares an obfuscated version of Y_j with others, denoted by $\hat{Y}_j$. In this chapter, it is assumed that each entity j will misreport its true observation results with probability $p_j^c \in [0, 0.5]$.[3] In addition, the entity with the best observation capability (e.g., highest detection rate in the intrusion detection case) is elected as the leader to make the final decision and all the other entities share their observations with the leader. In practice, the administrator of the network can work as the leader since it has a better view of the whole network (therefore a better observation capability), and has the authority to directly react to the attacks on behalf of the whole network. Once the attacker is identified and appropriate action is taken, the leader can share the information with the other entities at a later stage. In addition, the leader is assumed to be honest and will not deliberately lie about its knowledge of the attack.

3.1.3 An Overview of the Collaboration Model

Fig. 3.1 depicts the scenario in which there are 3 collaborative entities and a leader. In particular, the attacker chooses an attacking strategy π which determines the state of the network. Then, the entities observe the network state independently and obtain their observation results Y's. For the sake of privacy-preserving, each entity obfuscates its true results with a predetermined misreport probability and share the obfuscated observations $\hat{Y}$'s with the leader. Finally, given the shared observations from the collaborative entities, the leader can compute the probability distribution of the network state and therefore decide whether the attacker has launched an

[2]Note that the study of the follow-up moves (e.g., how to design a better attack against the entities with the security related information or how to deal with the possibly false data obtained during the attack) is highly non-trivial and deserves an independent work. Therefore, we focus on collaborative detection of the current attack in this chapter while leaving the follow-up moves of the attacker (and the investigation of the corresponding consequences) and the deception strategies of the entities as future works.

[3]In this chapter, it is assumed that the misreport probabilities are common knowledge for all the entities (i.e., they are shared together with the obfuscated observations). Therefore, it is equivalent for an entity to misreport with probability p_j^c or $1 - p_j^c$.

attack. In this sense, when the other entities have high privacy requirements and choose high misreport probabilities (e.g., 0.5), the leader barely obtains any useful information and can only make decision about the network state based on its own observation. However, when the entities care less about their privacy and share their observations with lower misreport probabilities, the leader gains more information and can estimate the network state more accurately. That being said, for the collaborative entities, a tradeoff between the collaboration gain and the preserved privacy exists. In addition, the main focus of this chapter is to study the benefit of collaboration, and therefore, it is first assumed that all the entities are obliged to collaborate by sharing their obfuscated observations and then it is shown that any rational entities will participate in the collaboration to maximize their utility.

3.2 Preliminaries of QIF

In this section, some basic concepts and properties of QIF are reviewed.

3.2.1 Quantitative Information Flow

We start by reviewing some basic notations in QIF. In particular, a *secret* is considered as something that is known to the adversary only as a prior probability distribution π. A *channel* $C : \mathcal{X} \times \mathcal{Y} \to \mathcal{R}$ is a function in which $\mathcal{X}$ is the set of input values and $\mathcal{Y}$ is the set of output values, and $C(x, y)$ measures the conditional probability of the channel output $y \in \mathcal{Y}$ when the input is $x \in \mathcal{X}$. It is typically assumed that the adversary knows the channel matrix C, which models the systems with observable behaviors that change the adversary's probabilistic knowledge, making the secret more vulnerable and hence causing information leakage [16]. Various vulnerability functions which measure the vulnerability of a secret and the information leakage caused by the observable behaviors of the systems exist. In this chapter, the secrets of the attacker and the entities, as well as some other notations, are defined as follows.

Definition 4. *The secret of the attacker (denoted by X) is whether it launches an attack or not (i.e., $X = 1$ if the attacker launches an attack and $X = 0$, otherwise).*[4] *The secrets of the entities are the observations they obtain, i.e., $Y_j = 0$ or $Y_j = 1, \forall j$.*

Remark 3. *Note that the secrets of the attacker and the entities are known deterministically to themselves but probabilistically to their opponents. When the attacker and the entities evaluate their vulnerability, they consider the amount of information that their opponents know about their secrets. Therefore, the vulnerabilities of the secrets are measured in the probabilistic sense.*

[4]Since the network state s is determined by whether an attack is launched, the attacker's secret is equivalently the true network state and $s = X$.

Definition 5. *The observation matrix Q^j for entity j is a matrix that satisfies $Q^j(x, y) = q_j(Y_j = y|s = x)$, in which $q_j(Y_j = y|s = x)$ is the probability of the private observation being $Y_j = y$ given the true network state $s = x$.*

Definition 6. *The obfuscation matrix R^j for entity j is a matrix that satisfies $R^j(y, \hat{y}) = p(\hat{Y}_j = \hat{y}|Y_j = y)$, in which $p(\hat{Y}_j = \hat{y}|Y_j = y)$ is the probability of the obfuscated observation being $\hat{Y}_j = \hat{y}$ given the true observation $Y_j = y$.*

Remark 4. *In this sense, the channel for the secret of entity j is the obfuscation matrix: $C(y_j, \hat{y}_j) = R^j(y_j, \hat{y}_j)$, which maps the true observation y_j to the obfuscated one $\hat{y}_j$; the channel for the secret of the attacker is given by $C(x, \hat{\boldsymbol{y}}) = \prod_j Q^j(x, y_j) \times R^j(y_j, \hat{y}_j)$, which maps the attacker's action x to the obfuscated observations of all the entities $\hat{\boldsymbol{y}} = [\hat{y}_1, \cdots, \hat{y}_N]$. Note that if entity k is elected as the leader, it does not have to obfuscate its observation and therefore $\hat{y}_k = y_k$.*

In addition, g-vulnerability [86], which is a general framework that can be adapted to many different vulnerability models, is adopted in our study. In the considered collaborative security framework, both the attacker and the collaborative entities aim to maximize their opponent's vulnerability and minimize their own. In the following, we take the attacker's secret as an example and introduce how to measure the vulnerability of the attacker using g-vulnerability. The vulnerability of the entities' secrets can be measured in a similar way.

g-vulnerability

In g-vulnerability, a gain function $g : \mathcal{W} \times \mathcal{X} \to R$, which measures the benefit of the leader (and therefore the entities) for making the guess $w \in \mathcal{W}$ when the secret of the attacker is $X = x \in \mathcal{X}$, is introduced. The g-vulnerability measures the threat to the attacker's secret as the entities choose the optimal guess w that maximizes its expected gain, and hence given the prior probability distribution π^A, the prior g-vulnerability is defined as

$$V_g(\pi^A) = sup_{w \in \mathcal{W}} \sum_{x \in \mathcal{X}} \pi_x^A g(w, x), \tag{3.1}$$

in which $\mathcal{W}$ is the set of possible guesses for the entities, and π_x^A is the prior probability of the secret being x, which is determined by the attacker.

By pushing the secret with prior probability distribution π^A to the channel C and observing the channel output $\hat{\boldsymbol{y}} \in \mathcal{Y}^N$, the entities can update their probabilistic knowledge on the attacker's secret. Given the prior probability distribution π^A and the channel C, the probability distribution of $\hat{\boldsymbol{y}}$ can be obtained. The posterior vulnerability is naturally defined as the expectation of the corresponding prior vulnerability (i.e., V_g) applied to each posterior distribution (i.e.,

$p(X|\hat{\boldsymbol{y}})$), weighted by the probability of each possible $\hat{\boldsymbol{y}}$ being realized. Therefore, the posterior g-vulnerability is given by [16]

$$\hat{V}_g(\pi^A, C) = \sum_{\hat{\boldsymbol{y}} \in \mathcal{Y}^N} p(\hat{\boldsymbol{y}}) V_g(p(X|\hat{\boldsymbol{y}})) = \sum_{\hat{\boldsymbol{y}} \in \mathcal{Y}^N} sup_{w \in \mathcal{W}} \sum_{x \in \mathcal{X}} \pi_x^A C(x, \hat{\boldsymbol{y}}) g(w, x), \tag{3.2}$$

in which $p(X|\hat{\boldsymbol{y}})$ is the posterior probability distribution of X given the output of channel $\hat{\boldsymbol{y}}$ and $p(\hat{\boldsymbol{y}})$ is the probability of $\hat{\boldsymbol{Y}} = \hat{\boldsymbol{y}}$; by the Bayes' formula, $p(X|\hat{\boldsymbol{y}}) = \frac{C(x,\hat{\boldsymbol{y}})\pi_x^A}{p(\hat{\boldsymbol{y}})}$; $C(x, \hat{\boldsymbol{y}})$ is given in Remark 4.

Finally, the information leakage L of the channel C is defined by the difference between the entities' prior knowledge and their posterior knowledge of the secret. The comparison is typically done additively, which is given by

$$L_g(\pi^A, C) = \hat{V}_g(\pi^A, C) - V_g(\pi^A). \tag{3.3}$$

Note that the g-vulnerability is a general framework due to the flexibility of selecting the gain functions. In the following, we further introduce one of the most commonly used vulnerability functions in this category [16].

Bayes-vulnerability

Bayes-vulnerability considers the entities trying to guess the secret of the attacker in one attempt and measures the threat to the attacker's secret as the probability of a correct guess [87]. Given the prior probability distribution π^A, the best strategy for the entities is to guess the secret to which it assigns the highest probability, and hence the Bayes-vulnerability is given by

$$V_b(\pi^A) = max_{x \in \mathcal{X}} \pi_x^A, \tag{3.4}$$

Similarly, the posterior Bayes-vulnerability is given by

$$\hat{V}_b(\pi^A, C) = \sum_{\hat{\boldsymbol{y}} \in \mathcal{Y}^N} p(\hat{\boldsymbol{y}}) V_b(p(X|\hat{\boldsymbol{y}})) = \sum_{\hat{\boldsymbol{y}} \in \mathcal{Y}^N} max_x \pi_x C(x, \hat{\boldsymbol{y}}). \tag{3.5}$$

By taking $\mathcal{W} = \mathcal{X}$ and let $g(w, x) = 1$ if and only if $w = x$ and 0 otherwise, one can easily verify that $V_g(\pi^A) = V_b(\pi^A)$. By taking proper gain function g, both Guessing-entropy [88] and Shannon-entropy [89], which are also commonly used as vulnerability measures, can be captured in this framework.

3.2.2 Properties of Posterior Vulnerabilities

In this subsection, some properties of posterior g-vulnerabilities are introduced as follows.

Definition 7. *An $n \times m$ channel C is useless [90] (or equivalently non-interfering [16]) if it has identical rows, i.e., $\sum_j C(i,j) = 1, \forall 1 \leq i \leq n$ and $C(i,j) = C(k,j), \forall 1 \leq i,k \leq n, 1 \leq j \leq m$.*

Lemma 1. ***Monotonicity (MONO)*** *[16]: Pushing a prior probability distribution through a channel C does not decrease vulnerability:*

$$\hat{V}_g(\pi, C) \geq V_g(\pi), \forall \pi, C, \tag{3.6}$$

in which the equality holds if and only if C is a useless channel or V_g is linear.

Lemma 2. ***Data-processing inequality (DPI)*** *[16]: Post-processing does not increase vulnerability:*

$$\hat{V}_g(\pi, C) \geq \hat{V}_g(\pi, CR), \forall \pi, C, R, \tag{3.7}$$

in which the number of columns in channel matrix C is the same as the number of rows in post-processing matrix R. The equality holds if and only if at least one of C and R is a useless channel or V_g is linear.

Lemma 3. ***Convexity (CVX)*** *[91]: Let $\{C_i\}_{i \in \mathcal{I}}$ be a family of channels that have the same input and output alphabets, and μ be a distribution on $\mathcal{I}$. Then for every prior distribution π, and every vulnerability V_g, the corresponding posterior vulnerability is convex with respect to channel composition, i.e., $\hat{V}_g(\pi, \sum_i \mu(i) C_i) \leq \sum_i \mu(i) \hat{V}_g(\pi, C_i)$.*

3.3 Quantitative Information Flow Game Model

In this section, three games are considered, each capturing one possible scenario of interest in practice. The first zero-sum game considers the scenario in which the entities work as a whole against the attacker. The second game generalizes the first zero-sum game into a general-sum game. The third game further considers independent and selfish entities. In this chapter, complete information games are considered (i.e., the utility functions and strategies are common knowledge), which is a common practice in literature [92–96]; incomplete games will be considered in future works.

In addition, g-vulnerability is adapted to measure the vulnerabilities of both the attacker's and the entities' secrets to capture more realistic settings.[5] Having the properties of posterior vulnerabilities at hand, the following propositions can be proved.

[5]In the following discussion, unless otherwise noted, the prior vulnerability and the posterior vulnerability refer to prior g-vulnerability and posterior g-vulnerability. However, in the examples and simulations, Bayes-vulnerability, as one special case of g-vulnerability, is adopted for the ease of illustration.

Proposition 1. *For any entity j that shares obfuscated observations with the leader, if neither Q^j nor R^j is a useless channel and V_g is nonlinear, the observation sharing process increases the vulnerabilities of both entity j's secret and the attacker's secret.*

Proof. Please see Appendix B.1. □

Proposition 2. *For any entity j that shares obfuscated observations with the leader, if neither Q^j nor R^j is a useless channel and V_g is nonlinear, the obfuscation process decreases the vulnerabilities of both the attacker's secret and entity j's secret. In particular, if the misreport probabilities of all the other entities are fixed, increasing the misreport probability of entity j decreases the vulnerabilities of both the attacker's secret and entity j's secret.*

Proof. Please see Appendix B.2. □

Remark 5. *Note that for the attacker's secret, $\hat{V}_g(\pi, C) = \sum_{\hat{\boldsymbol{y}} \in \mathcal{Y}^N} p(\hat{\boldsymbol{y}}) V_g(p_{X|\hat{\boldsymbol{y}}})$ $\geq V_g(\sum_{\hat{\boldsymbol{y}} \in \mathcal{Y}^N} p(\hat{\boldsymbol{y}}) p_{X|\hat{\boldsymbol{y}}}) = V_g(\pi)$, given that V_g is convex [16]. According to Jensen's inequality, the equality holds when $p_{X|\hat{\boldsymbol{y}}}$'s are the same for any $\hat{\boldsymbol{y}}$ or V_g is linear. Therefore, Proposition 1 and Proposition 2 hold only when V_g is nonlinear over $p_{X|\hat{\boldsymbol{y}}}, \forall \hat{\boldsymbol{y}}$. Otherwise, the observation sharing process and the obfuscation process (which determine $p_{X|\hat{\boldsymbol{y}}}$'s) do not change the posterior vulnerability of the attacker. Similar results can be verified for the entities.*

Note that Proposition 2 indicates that there is indeed a tradeoff between learning about the attacker's secret and keeping their own secrets for the collaborative entities. To investigate the optimal attacking strategy and obfuscation strategies for the attacker and entities, respectively, the following games are formulated and solved. For the ease of presentation, it is assumed that entity k is elected as the leader and the set of the other collaborative entities is denoted as $\mathcal{N}_c$. In particular, since the leader does not have to share anything with others, the tradeoff of the collaboration gain and privacy loss is considered only for the entities in $\mathcal{N}_c$.

3.3.1 Game I: Zero-Sum Fully Collaborative Game

In this game, it is assumed that the entities always act as a whole, which means that each entity takes into consideration not only its own privacy but also that of the whole network. In addition, it is assumed without loss of generality that the action space of both the attacker and the entities is finite and denoted by $\mathcal{A} = \{\pi_L^A, \pi_H^A\}$ and $\mathcal{D} = \{R_L, R_H\}$, respectively.[6] π_L^A and π_H^A correspond to attacking with low probability (i.e., low $\pi^A(a_1)$) and high probability (i.e., high $\pi^A(a_1)$) for the attacker, respectively, while R_L and R_H are the channel matrices that correspond to

[6]The study can be extended to the scenarios with larger action space.

misreporting with low probability and high probability for the entities, respectively. Note that for any $R \in \mathcal{D}$,

$$R(\boldsymbol{Y}, \hat{\boldsymbol{Y}}) = \prod_{j=1}^{N} R^j(Y_j, \hat{Y}_j), \tag{3.8}$$

in which $\boldsymbol{Y} = [Y_1, \cdots, Y_N]$ is the true observations of all the entities and $\hat{\boldsymbol{Y}} = [\hat{Y}_1, \cdots, \hat{Y}_{k-1}, Y_k, \hat{Y}_{k+1}, \cdots, \hat{Y}_N]$ consists of the received obfuscated observations from the entities and the leader's true observation; R^j is the obfuscation matrix of entity j.[7] In the case that the attacker chooses action $\pi^A \in \mathcal{A}$ and the entities choose action $R \in \mathcal{D}$, the reward of the attacker is given by

$$u_A(\pi^A, R) = \lambda_A \sum_{j \in \mathcal{N}_c} \hat{V}_{Y_j}(\pi^A, R) - \hat{V}_X(\pi^A, R), \tag{3.9}$$

in which $\hat{V}_{Y_j}(\pi^A, R)$ is the posterior vulnerability of the secret of entity j (i.e., Y_j) given the obfuscation matrix R; $\hat{V}_X(\pi^A, R)$ is the posterior vulnerability of the secret of the attacker (i.e., X) given the obfuscation matrix R; λ_A is the importance factor of learning the entities' secrets for the attacker. Note that instead of information leakage, the posterior vulnerability is used to model the rewards of both the attacker and the entities, because it measures how vulnerable their secrets are to the opponents and therefore determines their security performance. In addition, due to the zero-sum setting, the rewards of the entities are given by

$$u_E(\pi^A, R) = -u_A(\pi^A, R). \tag{3.10}$$

Note that $\hat{V}_{Y_j}(\pi^A, R)$ and $\hat{V}_X(\pi^A, R)$ depend on the vulnerability functions used to measure the posterior vulnerability of the secrets of the entities and the attacker, respectively. When g-vulnerability is adopted, they are given by

$$\hat{V}_{Y_j}(\pi^A, R) = \sum_X \pi^A(X) \hat{V}_{Y_j|X}(\pi^A, R) = \sum_X \pi^A(X) \sum_{\hat{Y}_j} sup_w \sum_{Y_j} Q^j(X, Y_j) p(\hat{Y}_j|Y_j) g(w, Y_j), \tag{3.11}$$

$$\hat{V}_X(\pi^A, R) = \sum_{\hat{\boldsymbol{Y}}} sup_w \sum_X \pi^A(X) p(\hat{\boldsymbol{Y}}|X) g(w, X), \tag{3.12}$$

in which $\pi^A(X)$ measures the probability that the true network state is X; $\hat{V}_{Y_j|X}(\pi^A, R)$ measures the posterior vulnerability of entity j's secret (from the attacker's view) given the true network state X;[8] $Q^j(X, Y_j)$ is the probability that the true observation of entity j is Y_j given that the

[7] Note that R^j may be different for different j, so the entities may have different misreport probabilities.

[8] Note that after the attacker takes an action, it can estimate $\hat{V}_{Y_j|X}$ since it knows the true network state X. However, before the attacker determines its attacking strategy (i.e., $\pi^A(X)$) and takes an action, X is unknown to anyone and therefore the vulnerability of entity j's secret is modeled as the average posterior vulnerability over

true network state is X; $p(\hat{Y}_j|Y_j)$ measures the probability that entity j shares $\hat{Y}_j$ given the true observation Y_j; $p(\hat{\boldsymbol{Y}}|X)$ measures the probability that the entities share $\hat{\boldsymbol{Y}}$ when the true network state is X.[9]

Therefore, the expected utility functions of both the attacker and the entities are given by

$$U_A(p_A(\pi^A), p_E(R)) = \sum_{\pi^A \in \mathcal{A}} \sum_{R \in \mathcal{D}} p_A(\pi^A) p_E(R) u_A(\pi^A, R), \tag{3.13}$$

$$U_E(p_A(\pi^A), p_E(R)) = -U_A(p_A(\pi^A), p_E(R)), \tag{3.14}$$

in which $p_A(\pi^A)$ is the probability that the attacker takes action π^A and $p_E(R)$ is the probability that the entities take action R.

Theorem 5. *(von Neumann's minimax theorem). Let $\mathcal{X} \subset \mathcal{R}^m$ and $\mathcal{Y} \subset \mathcal{R}^n$ be compact sets, and $\mathcal{U} : \mathcal{X} \times \mathcal{Y} \to \mathcal{R}$ be a continuous function such that $U(x, y)$ is a convex function in $x \in \mathcal{X}$ and a concave function in $y \in \mathcal{Y}$. Then:*

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} U(x, y) = \max_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} U(x, y). \tag{3.15}$$

Corollary 2. *The Nash Equilibrium (NE) of Game I is given by*

$$p_E^*(R_L) = \frac{u_E(\pi_H^A, R_H) - u_E(\pi_L^A, R_H)}{u_E(\pi_L^A, R_L) - u_E(\pi_H^A, R_L) - u_E(\pi_L^A, R_H) + u_E(\pi_H^A, R_H)}, \tag{3.16}$$

$$p_A^*(\pi_L^A) = \frac{u_E(\pi_H^A, R_H) - u_E(\pi_H^A, R_L)}{u_E(\pi_L^A, R_L) - u_E(\pi_H^A, R_L) - u_E(\pi_L^A, R_H) + u_E(\pi_H^A, R_H)}, \tag{3.17}$$

if these values are in $[0, 1]$, in which $p_E^*(R_L)$ is the probability that the entities take action R_L and $p_A^*(\pi_L^A)$ is the probability that the attacker take action π_L^A. Otherwise, there exist pure strategies. In particular, when

$$\frac{u_E(\pi_H^A, R_H) - u_E(\pi_H^A, R_L)}{u_E(\pi_L^A, R_L) - u_E(\pi_H^A, R_L) - u_E(\pi_L^A, R_H) + u_E(\pi_H^A, R_H)} \notin [0, 1], \tag{3.18}$$

the entities have pure strategy

$$p_E^*(R_L) = \begin{cases} 1, \text{ if } u_E(\pi_H^A, R_H) - u_E(\pi_H^A, R_L) < 0, \\ 0, \text{ if } u_E(\pi_H^A, R_H) - u_E(\pi_H^A, R_L) > 0. \end{cases} \tag{3.19}$$

$\pi^A(X)$.

[9]Note that $p(\hat{Y}_j|Y_j)$ is determined by the obfuscation matrix R and $p(\hat{\boldsymbol{Y}}|X)$ is determined by observation matrices Q's and the obfuscation matrix R.

Therefore, the attacker has pure strategy

$$\pi^A = \underset{\pi \in \mathcal{A}}{\operatorname{argmax}}\, u_A(\pi, R), \tag{3.20}$$

in which $R \in \mathcal{D}$ is determined by (3.19). Similarly, when

$$\frac{u_E(\pi_H^A, R_H) - u_E(\pi_L^A, R_H)}{u_E(\pi_L^A, R_L) - u_E(\pi_H^A, R_L) - u_E(\pi_L^A, R_H) + u_E(\pi_H^A, R_H)} \notin [0, 1], \tag{3.21}$$

the attacker has pure strategy

$$p_A^*(\pi_L^A) = \begin{cases} 1, \text{ if } u_E(\pi_H^A, R_H) - u_E(\pi_L^A, R_H) > 0, \\ 0, \text{ if } u_E(\pi_H^A, R_H) - u_E(\pi_L^A, R_H) < 0. \end{cases} \tag{3.22}$$

Therefore, the entities have pure strategy

$$R = \underset{R \in \mathcal{D}}{\operatorname{argmax}}\, u_A(\pi^A, R), \tag{3.23}$$

in which $\pi^A \in \mathcal{A}$ is determined by (3.22).

Proof. Please see Appendix B.3. □

Example 1. *Consider a collaborative security system consisting of 2 entities that observe the network state and collaborate by sharing obfuscated observations with a leader. Assume that both the two entities and the leader can observe the network state correctly with probability 0.95 (0.7) when $s = 0$ ($s = 1$). In addition, the attacker has two possible attacking strategies with $\pi_L^A = 0.1$ and $\pi_H^A = 0.6$. The entities act as a whole and they have two possible collaboration strategies. In particular, when they choose R_H, one entity misreports with probability 0.2 and the other one misreports with probability 0.5; when they choose R_L, one entity misreports with probability 0.1 and the other one misreports with probability 0.2. Finally, both the attacker and the entities adopt Bayes-vulnerability to measure their vulnerability. Therefore, the utility function of the attacker is given by*

$$u_A(\pi^A, R) = \begin{cases} 1.88\lambda_A - 0.7208, \text{ for } (\pi^A, R) = (\pi_L^A, R_L), \\ 1.86\lambda_A - 0.7080, \text{ for } (\pi^A, R) = (\pi_L^A, R_H), \\ 1.78\lambda_A - 0.7033, \text{ for } (\pi^A, R) = (\pi_H^A, R_L), \\ 1.66\lambda_A - 0.7, \text{ for } (\pi^A, R) = (\pi_H^A, R_H), \end{cases} \tag{3.24}$$

in which λ_A is the importance factor of learning the entities' secrets for the attacker. As a result, the payoff matrix of Game I is given by Table 3.2.

Table 3.2: Payoff matrix of Game I

	R_L	R_H
π_L^A	$1.88\lambda_A - 0.7208$, $-1.88\lambda_A + 0.7208$	$1.86\lambda_A - 0.7080$, $-1.86\lambda_A + 0.7080$
π_H^A	$1.78\lambda_A - 0.7033$, $-1.78\lambda_A + 0.7033$	$1.66\lambda_A - 0.7$, $-1.66\lambda_A + 0.7$

According to Corollary 2, the NE is given by

$$p_E^*(R_L) = \frac{0.2\lambda_A - 0.008}{0.1\lambda_A + 0.0095}, p_A^*(\pi_L^A) = \frac{0.12\lambda_A - 0.0033}{0.1\lambda_A + 0.0095},$$

if these values are in $[0, 1]$*. Otherwise, there exist pure strategies as suggested in Corollary 2. In particular, when* λ_A *is chosen such that* $\frac{0.12\lambda_A - 0.0033}{0.1\lambda_A + 0.0095} \notin [0, 1]$*, we have* $p_E^*(R_L) = 0$ *if* λ_A *is large (e.g.,* $\lambda_A = \infty$*) and* $p_E^*(R_L) = 1$ *if* λ_A *is small (e.g.,* $\lambda_A = 0$*).*[10]

Remark 6. *From the above example, it can be observed that for the same attacking strategy, when the entities switch from* R_L *to* R_H*, the vulnerabilities of both the attacker and the entities decrease. This is because, when the entities misreport with higher probability, the shared obfuscated observations contain less information about both the attacker's secret (i.e., whether it has launched an attack or not) and the entities' secrets (i.e., the true observation results). Therefore, for the entities, there is indeed a tradeoff between the collaboration gain and the privacy loss, and the optimal collaboration strategy depends on the importance factor* λ_A*. In particular, when* λ_A *is large (i.e., the attacker allocates more importance on learning the entities' secrets or equivalently the entities allocate more importance on preserving their own secrets), the entities tend to obfuscate their observations with a high probability. In contrast, when* λ_A *is small, the entities tend to obfuscate their observations with a low probability.*

3.3.2 Game II: Non-zero-sum Fully Collaborative Game

In Game I, the problem is modeled as a zero-sum game, in which the defender's utility is exactly the opposite of that of the attacker. In practice, the QIF game may not necessarily be zero-sum and need to be generalized to capture the non-zero-sum setting.[11] For instance, when the entities and the attacker have different importance factors of learning each other's secret (i.e., λ_A in

[10]Note that the utility of the entities is not necessarily positive. For instance, in practice there are often constraints in the detection performance, i.e., the posterior vulnerability of the attacker must be higher than a threshold for security guarantee. In this case, the entities may be required to misreport with a probability of less than 0.5 even if they select R_L. Consequently, a large λ_A may result in negative utility for the entities, which can be viewed as the price paid for higher security.

[11]Note that the following results are obtained from the perspective of the collaborative entities. By switching the roles of the attacker and the entities, one can derive similar results from the perspective of the attacker.

(3.9)), the utility function of the entities may differ from (3.10). With such consideration, the rewards of the attacker and the entities in this game are modeled by

$$u_A(\pi^A, R) = \lambda_A \sum_{j \in \mathcal{N}_c} \hat{V}_{Y_j}(\pi^A, R) - \hat{V}_X(\pi^A, R), \tag{3.25}$$

$$u_E(\pi^A, R) = -\lambda_E \sum_{j \in \mathcal{N}_c} \hat{V}_{Y_j}(\pi^A, R) + \hat{V}_X(\pi^A, R), \tag{3.26}$$

in which λ_E is the importance factor of preserving their secrets for the entities.

Similarly, the utility functions of both the attacker and the entities are given by

$$U_A(p_A(\pi^A), p_E(R)) = \sum_{\pi^A \in \mathcal{A}} \sum_{R \in \mathcal{D}} p_A(\pi^A) p_E(R) u_A(\pi^A, R), \tag{3.27}$$

$$U_E(p_A(\pi^A), p_E(R)) = \sum_{\pi^A \in \mathcal{A}} \sum_{R \in \mathcal{D}} p_A(\pi^A) p_E(R) u_E(\pi^A, R). \tag{3.28}$$

Note that due to the non-zero-sum formulation, many properties of Game I no longer hold in Game II. For example, a minimax strategy in Game II may not be a maximin strategy, which means that we can no longer use Theorem 5 to obtain the NE of Game II. In this case, the NE strategies are determined by the importance factors λ_E and λ_A.

Case 1:

When $\lambda_E \leq \lambda_{min} \triangleq \min(\frac{\hat{V}_X(\pi_L^A, R_H) - \hat{V}_X(\pi_L^A, R_L)}{L_Y(\pi_L^A, R_H) - L_Y(\pi_L^A, R_L)}, \frac{\hat{V}_X(\pi_H^A, R_H) - \hat{V}_X(\pi_H^A, R_L)}{L_Y(\pi_H^A, R_H) - L_Y(\pi_H^A, R_L)})$, where $L_Y(\pi^A, R) = \sum_{j \in \mathcal{N}_c} \hat{V}_{Y_j}(\pi^A, R)$, it can be verified that

$$u_E(\pi_L^A, R_L) > u_E(\pi_L^A, R_H), \tag{3.29}$$

$$u_E(\pi_H^A, R_L) > u_E(\pi_H^A, R_H). \tag{3.30}$$

Therefore, R_L is a dominant strategy for the entities. In this case, there exists a pure NE strategy given by

$$R = R_L; \pi^A = \underset{\pi \in \mathcal{A}}{\operatorname{argmax}}\, u_A(\pi, R_L).$$

Case 2:

When $\lambda_E \geq \lambda_{max} \triangleq \max(\frac{\hat{V}_X(\pi_L^A, R_H) - \hat{V}_X(\pi_L^A, R_L)}{L_Y(\pi_L^A, R_H) - L_Y(\pi_L^A, R_L)}, \frac{\hat{V}_X(\pi_H^A, R_H) - \hat{V}_X(\pi_H^A, R_L)}{L_Y(\pi_H^A, R_H) - L_Y(\pi_H^A, R_L)})$, it can be verified that

$$u_E(\pi_L^A, R_L) < u_E(\pi_L^A, R_H), \tag{3.31}$$

$$u_E(\pi_H^A, R_L) < u_E(\pi_H^A, R_H). \tag{3.32}$$

Therefore, R_H is a dominant strategy for the entities. In this case, there exists a pure NE strategy given by

$$R = R_H; \pi^A = \underset{\pi \in \mathcal{A}}{\operatorname{argmax}}\, u_A(\pi, R_H).$$

Case 3:

When $\lambda_E \in (\lambda_{min}, \lambda_{max})$, there is no dominant strategy for the entities, and hence mixed NE strategies are explored. Particularly, we are interested in the case that $\lambda_A \in (\lambda_{min}, \lambda_{max})$. In this case, when the entities switch to another pure strategy (e.g., from R_L to R_H) that increases their payoffs, the payoff of the attacker decreases, which means that the attacker and the entities are competing to maximize their own utility while minimizing their opponent's. Using similar technique as that in [97], it can be proved that any entity's NE strategy is also a minimax strategy.

Define the set of entities' minimax strategies as:

$$\Omega_{Minimax} = \underset{p_E}{\operatorname{argmin}} \max_{p_A} U_A(p_A, p_E). \tag{3.33}$$

For any attacker's strategy $p_A(\pi^A) = [p_A(\pi_L^A), p_A(\pi_H^A)]$, define a function $f(p_A(\pi^A)) = \hat{p}_A(\pi^A)$ such that

$$\hat{p}_A(\pi_L^A) = \eta \frac{u_E(\pi_L^A, R_L) - u_E(\pi_L^A, R_H)}{u_A(\pi_L^A, R_H) - u_A(\pi_L^A, R_L)} p_A(\pi_L^A), \tag{3.34}$$

$$\hat{p}_A(\pi_H^A) = \eta \frac{u_E(\pi_H^A, R_L) - u_E(\pi_H^A, R_H)}{u_A(\pi_H^A, R_H) - u_A(\pi_H^A, R_L)} p_A(\pi_H^A), \tag{3.35}$$

where $\eta > 0$ is chosen to satisfy $\hat{p}_A(\pi_L^A) + \hat{p}_A(\pi_H^A) = 1$. Then the following Lemma holds.

Lemma 4. *When $\lambda_E, \lambda_A \in (\lambda_{min}, \lambda_{max})$, (p_A, p_E) is an NE of Game II if and only if $(f(p_A), p_E)$ is an NE in Game I.*

Proof. Please see Appendix B.4. □

Theorem 6. *When $\lambda_E, \lambda_A \in (\lambda_{min}, \lambda_{max})$, the set of entities' minimax strategies is identical to the set of entities' NE strategies in Game II.*

Proof. Please see Appendix B.5. □

To this end, we can solve Game II by first obtaining the entities' minimax strategy, based on which the corresponding NE strategy of the attacker can be further derived.

Remark 7. *Note that in Game I and Game II, the action sets of the entities are determined by the whole network. For those entities that are not willing to collaborate, their misreport probabilities can be set as 0.5 so that there is no privacy leakage for them. In this case, all the entities can obtain higher reward by following the leader's decision when the misreport probability of any entity is less than 0.5, since the leader has the best observation capability and according to Proposition 1, receiving obfuscated observations from the entities further increases the attacker's posterior vulnerability (i.e., larger $\hat{V}_X$) from the leader's perspective. Therefore, all the entities will participate in the collaboration and act as a whole.*

3.3.3 Game III: Two-layer Non-fully Collaborative Game

Different from Game I/II, a scenario in which the entities assume certain independence is considered in this game. In particular, the entities are willing to cooperate for the joint benefits temporarily but their ultimate concern is their own privacy. In addition, larger action sets for the attacker and the entities are assumed, denoted by $\mathcal{A} = \{\pi_1^A, \pi_2^A, \cdots, \pi_{M_A}^A\}$ and $\mathcal{P}^{mis} = \{p^1, p^2, \cdots, p^{M_D}\}$, with $p^i \in [0, 0.5], \forall 1 \leq i \leq M_D$ and $\pi_j^A \in [0, 1], \forall 1 \leq j \leq M_A$, respectively. With these considerations, the problem is modeled as a two-layer single-leader multi-follower game, in which the attacker acts as the leader and entities act as the followers that are informed of the attacker's attacking strategy. The first-layer game models the interaction between the attacker and the entities, while the second-layer game models the collaborative information sharing among the entities themselves.

In this case, the utility function of the attacker is given by[12]

$$U_A(\pi^A, \boldsymbol{p^c}) = \sum_{j \in \mathcal{N}_c} \lambda_j \hat{V}_{Y_j}(\pi^A, p_j^c) - \hat{V}_X(\pi^A, \boldsymbol{p^c}), \tag{3.36}$$

in which $\boldsymbol{p^c} = [p_1^c, p_2^c, \cdots, p_N^c]$ is a vector consisting of the entities' misreport probabilities and $p_j^c \in \mathcal{P}^{mis}, \forall j \in \mathcal{N}$; $\hat{V}_{Y_j}(\pi^A, p_j^c)$ is the posterior vulnerability of entity j's secret given its misreport probability; $\hat{V}_X(\pi^A, \boldsymbol{p^c})$ is the posterior vulnerability of the attacker's secret given the misreport probabilities of the entities; λ_j is the importance factor of learning Y_j for the attacker. Similarly, the utility function of entity j is given by

$$U_{E_j}(\pi^A, \boldsymbol{p^c}) = \hat{V}_X(\pi^A, \boldsymbol{p^c}) - \lambda_j \hat{V}_{Y_j}(\pi^A, p_j^c). \tag{3.37}$$

Note that the leader-follower game is often solved by backward induction. First, solve the follower's problem for every possible strategy taken by the leader. The solution consists of the

[12]In this case, the utility of the attacker may be negative even when it chooses not to attack. However, since the utility function as well as the vulnerability function (i.e., the gain function $g(w, x)$ in g-vulnerability) are subject to design, one can set the utility of such outside options for the attacker as 0.

best response strategy of the follower as a function of the leader's strategy. Then, the leader decides its optimal strategy according to the follower's best responses. Therefore, for every possible π^A taken by the attacker, we first solve the second-layer game among the collaborative entities.

Theorem 7. *[98] Every finite potential game admits at least one pure-strategy NE.*

Corollary 3. *For any attacking strategy π^A, the second-layer game in Game III is a potential game and therefore admits at least one pure-strategy NE.*

Proof. Consider the function

$$P(\pi^A, \boldsymbol{p^c}) = \hat{V}_X(\pi^A, \boldsymbol{p^c}) - \sum_{j \in \mathcal{N}_c} \lambda_j \hat{V}_{Y_j}(\pi^A, p_j^c). \tag{3.38}$$

It can be shown that

$$P(\pi^A, [p_j^c, \boldsymbol{p^c_{-j}}]) - P(\pi^A, [p_j'^c, \boldsymbol{p^c_{-j}}]) = U_{E_j}(\pi^A, [p_j^c, \boldsymbol{p^c_{-j}}]) - U_{E_j}(\pi^A, [p_j'^c, \boldsymbol{p^c_{-j}}]), \\ \forall p_j'^c, p_j^c, \pi^A \in [0,1], j \in \mathcal{N}, \boldsymbol{p^c_{-j}} \in [0,1]^{N-1}. \tag{3.39}$$

Therefore, P is a potential function for the second-layer game in Game III, which means it is a potential game. As a result, according to Theorem 7, the second-layer game in Game III admits at least one pure-strategy NE. □

Given the existence of NE at hand, the log-linear learning algorithm [99] can be adapted to compute the NE.

Theorem 8. *[99] With a sufficiently large β, the proposed log-linear learning algorithm asymptotically converges to an action profile that maximizes the potential function.*

Note that the utility functions of all the entities are publicly known and therefore an entity can mimic the behaviors of the others and run Algorithm 6 locally. To this end, for every possible mixed strategy taken by the attacker, the entities can learn their optimal collaboration strategies by running Algorithm 6, which maximizes the potential function $P(\pi^A, \boldsymbol{p}^c)$ given by (3.38). Since $P(\pi^A, \boldsymbol{p}^c) = -U_A(\pi^A, \boldsymbol{p}^c)$, the obtained NE strategy is indeed the optimal one which minimizes the attacker's utility. In the meantime, by running the same algorithm, the attacker can infer the optimal collaboration strategies of the entities and therefore find its optimal attacking strategy.

Theorem 9. *For any attacking strategy π^A from the attacker, a rational entity that intends to maximize its own utility will choose to participate in the collaboration.*

Proof. Note that the utility function of entity j in the collaborative case (at NE) is given by $U_{E_j}^{NE}(\pi^A, \boldsymbol{p^c}) = \hat{V}_X(\pi^A, \boldsymbol{p^c}) - \lambda_j \hat{V}_{Y_j}(\pi^A, p_j^c)$. If entity j chooses not to collaborate and

Algorithm 6 Log-linear Learning Algorithm

Initialization: Set the iteration index $t = 0$, let each collaborative entity $j, \forall j \in \mathcal{N}_c$ randomly select a misreport probability $p_j^c \in \mathcal{P}^{mis}$ and set the binary flag $x_j(t) = 0, \forall j \in \mathcal{N}_c$. All the collaborative entities simultaneously execute the following procedure:

for $t = 1, 2, ...$ **do**

Exploration:

if $x_j(t-1) = 0$ **then**

Entity j updates its selection according to the following rule:

$$Pr[p_j^c(t) = a] = \begin{cases} \dfrac{\gamma_j}{|\mathcal{P}^{mis}| - 1}, a \in \{\mathcal{P}^{mis} \backslash p_j^c(t-1)\}, \\ 1 - \gamma_j, \qquad\qquad a = p_j^c(t-1), \end{cases}$$

in which $|\mathcal{P}^{mis}|$ is the cardinality of the set $\mathcal{P}^{mis}$ and γ_j is the exploration rate of entity j. In addition, set $x_j(t) = 1$ if $p_j^c(t) \neq p_j^c(t-1)$, and $x_j(t) = 0$ otherwise.

end if

Update:

if $x_j(t-1) = 1$ **then**

Entity j updates the selection according to the following rule:

$$\begin{aligned} Pr[p_j^c(t) = p_j^c(t-1)] &= \frac{e^{u_{E_j}(t-1)\beta}}{X}, \\ Pr[p_j^c(t) = p_j^c(t-2)] &= \frac{e^{u_{E_j}(t-2)\beta}}{X}, \end{aligned} \tag{3.40}$$

in which β is the learning parameter, $u_{E_j}(t-1)$ and $u_{E_j}(t-2)$ are the received utility function of entity j in iteration $t-1$ and $t-2$, respectively, and $X = e^{u_{E_j}(t-1)\beta} + e^{u_{E_j}(t-2)\beta}$. Furthermore, set $x_j(t)$=0.

end if

end for

makes decision on whether the attacker has launched attacks based on its own observation, the posterior vulnerability of the attacker is given by $\hat{V}_X^{Non}(\pi^A, Q^j)$ and its own posterior vulnerability is equal to its prior vulnerability $\hat{V}_{Y_j}^{Non}(\pi^A) = V_{Y_j}(\pi^A)$. Since the leader (i.e., entity k) has the best observation capability among all the entities, it is immediate that $\hat{V}_X^{Non}(\pi^A, Q^j) \leq \hat{V}_X^{Non}(\pi^A, Q^k)$. Furthermore, when all the other entities choose the same collaborative strategies as those at NE, according to Proposition 1, the observation sharing process increases the posterior vulnerability of the attacker from the leader's view, and therefore $\hat{V}_X^{Non}(\pi^A, Q^k) \leq \hat{V}_X(\pi^A, [p_j^c = 0.5, \boldsymbol{p}_{-j}^{\boldsymbol{c}}])$. On the other hand, as is discussed in Remark 5, $\hat{V}_{Y_j}(\pi^A, p_j^c = 0.5) = V_{Y_j}(\pi^A)$. As a result, the utility of the non-collaborative entity j is $U_{E_j}^{Non}(\pi^A) = \hat{V}_X^{Non}(\pi^A, Q^j) - \lambda_j \hat{V}_{Y_j}^{Non}(\pi^A) \leq U_{E_j}^{NE}(\pi^A, [p_j^c = 0.5, \boldsymbol{p}_{-j}^{\boldsymbol{c}}])$. In addition, due to the

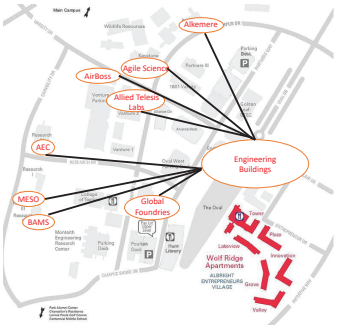

Figure 3.2: NCSU Centennial Campus Map

property of NE, the utility of entity j at NE satisfies $U_{E_j}^{NE}(\pi^A, [p_j^c = 0.5, \boldsymbol{p}_{-j}^{\boldsymbol{c}}]) \leq U_{E_j}^{NE}(\pi^A, \boldsymbol{p}^{\boldsymbol{c}})$. Therefore,

$$U_{E_j}^{Non}(\pi^A) \leq U_{E_j}^{NE}(\pi^A, [p_j^c = 0.5, \boldsymbol{p}_{-j}^{\boldsymbol{c}}]) \leq U_{E_j}^{NE}(\pi^A, \boldsymbol{p}^{\boldsymbol{c}}). \tag{3.41}$$

□

Remark 8. *Note that in (3.41), the first equality holds only when either of the two conditions is satisfied. (1) Entity j has the same observation capability as the leader and the misreport probabilities of all the other entities that participate in the collaboration are 0.5 at NE. (2) The attacker acts deterministically, i.e., it always chooses to launch (not to launch) attack. The second equality holds when the misreport probability of entity j is 0.5 at NE if it chooses to collaborate. These two conditions correspond to extreme events that in general are not of interest in practice. In any case, participating in collaboration will incur no loss for the entities.*

3.4 Numerical Illustrations

In this section, we apply our game-theoretic analysis to the case of collaborative intrusion detection against an attacker in the centennial campus of North Carolina State University (NCSU). In the centennial campus, NCSU shares the same network with its industrial partners as shown in Fig. 3.2. We consider a scenario in which the network faces a smart attacker and the companies collaborate by sharing their observations with an administrator, which resides in the Engineering Buildings and further determines whether there is an attack or not. In this case, the administrator in the Engineering Buildings works as the leader while all the 8 companies work as the entities in our model. In the simulation results presented in this section, the participated companies are randomly selected from these 8 companies with $1 \leq |\mathcal{N}_c| \leq 8$ and $N = |\mathcal{N}_c| + 1$. An ideal scenario, in which the shared information can be collected by the leader without any delay or errors, is considered. Since the NE strategies of both the attacker and entities in Game

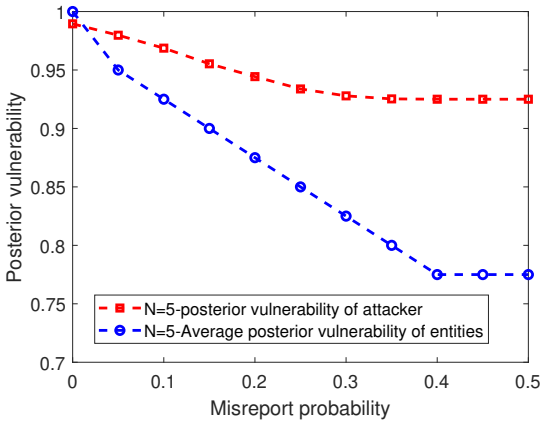

Figure 3.3: Collaboration utility-Privacy tradeoff curve.

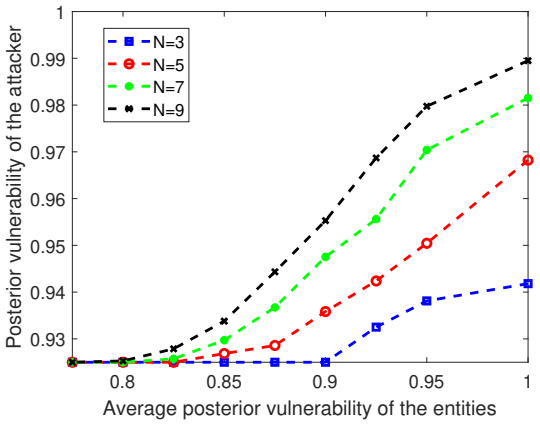

Figure 3.4: Posterior vulnerability vs misreport probability.

I and Game II can be theoretically derived, we mainly present the numerical results for Game III in this section.

3.4.1 Utility-privacy Tradeoff

In this subsection, the tradeoff between the collaboration utility (i.e., $\hat{V}_X$) and the privacy (i.e., $\frac{1}{N-1}\sum_{j\in\mathcal{N}_c}\hat{V}_{Y_j}$) is examined, given that $N-1$ entities (i.e., companies) participate in the collaboration. In particular, instead of focusing on the NE strategies in the proposed game models, the posterior vulnerabilities of both the attacker and the entities for all possible misreport probabilities are examined. It is assumed that the entities can observe the network state correctly with probability 0.6 (0.95) when the attacker launches (does not launch) an attack, while the leader can observe the network state correctly with probability 0.9 (0.95) when the attacker launches (does not launch) an attack. In addition, Bayes-vulnerability is used to measure the vulnerability of both the attacker and the entities.

Fig. 3.3 shows the relationship between the misreport probabilities of the entities and the corresponding posterior vulnerability of both the entities and the attacker, in the case that $N = 5$ and the attacker attacks with probability 0.5. In addition, it is assumed that all the entities misreport with the same probability.[13] It can be seen that for any misreport probability $p^c < 0.5$, the posterior vulnerability of the attacker is no smaller than 0.9 while that of the entities is larger than 0.775, which means the observation sharing process indeed leads to larger posterior vulnerabilities and therefore information leakage for the collaborative entities, and at

[13]Similar results are obtained when the entities adopt different misreport probabilities and/or the attacker adopts other attacking strategies, which are omitted in the interest of space.

the same time may render the attacker more vulnerable.[14] In addition, compared to the case that the entities share their observations honestly ($p^c = 0$), both the posterior vulnerability of the attacker and the average posterior vulnerability of the entities decrease as the misreport probability increases, which indicates that the obfuscation process decreases the vulnerabilities of both the attacker's and the entities' secrets.

Fig. 3.4 shows the tradeoff between the posterior vulnerability of the attacker and the average posterior vulnerability of the entities. In general, larger average posterior vulnerability for the entities corresponds to larger posterior vulnerability for the attacker. This is because when the entities misreport with lower probability, the shared observations are more accurate and contain more information about the true network state, which helps the leader estimate the attacker's action more accurately. In addition, it can be seen that with more collaborative entities, the posterior vulnerability of the attacker in most cases increases for the same average posterior vulnerability of the entities. Intuitively, when there are more collaborative entities, the leader can gather more information about whether the attacker launches an attack or not and obtain a better estimate.

In addition, it can be observed that when the misreport probability is large (equivalently the average posterior vulnerability of the entities is small), the posterior vulnerabilities of both the attacker and the entities stop decreasing as the misreport probability increases. This can be explained as follows: when the misreport probability is high, the Bayes-vulnerability function becomes linear. According to the discussion in Remark 5, the posterior vulnerability of the attacker does not change as p^c increases.

3.4.2 Convergence of the Log-linear Learning Algorithm

In this subsection, the convergence of the log-linear learning algorithm is examined. Similar to the setting in Section 3.4.1, a network consisting of N collaborative entities is considered. It is assumed that both the entities and the leader can observe the network state correctly with probability 0.7 (0.95) when the attacker launches (does not launch) an attack. The λ's (i.e., the importance factor of learning the attacker's secret) are set to 0.05 for all the entities. In addition, the action sets for the entities are set as $\mathcal{P}^{mis} = \{0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$; the exploration rate used in the log-linear algorithm is chosen as $\gamma_j = e^{-10^{-5}\beta t}$ in which $\beta = 500$ and t is the iteration index, for all j. Fig. 3.5 shows the average reward of the entities of the second-layer game in Game III in the case that the attacker takes the attacking strategy $\pi^A = 0.5$. It can be seen that for all the scenarios examined, the average reward of the entities converges.

[14] Note that in this case, given the observation of the leader, the posterior vulnerability of the attacker is 0.9, while the average prior vulnerability of the entities is equal to 0.775.

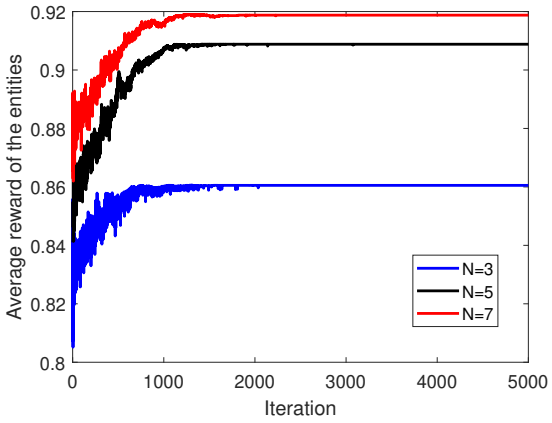

Figure 3.5: The convergence of the log-linear learning algorithm

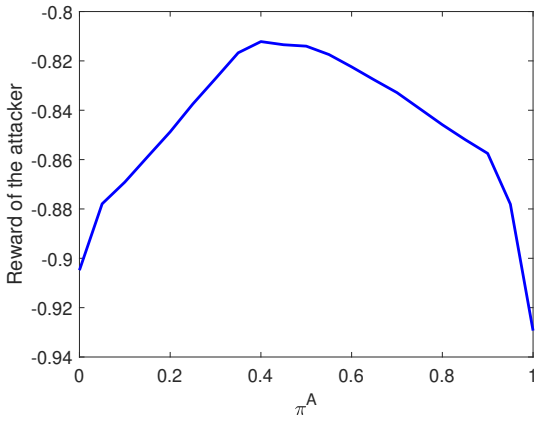

Figure 3.6: Optimal attacking strategy

3.4.3 Optimal Attacking Strategy

In this subsection, the optimal attacking strategy is investigated. In particular, a network consisting of 3 collaborative entities is considered. The action set of the attacker is set as $\mathcal{A} = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ and the other parameters are the same as those in Section 3.4.2. Fig. 3.6 shows the attacker's reward in terms of attacking probability. It can be seen that in this case, the optimal attacking strategy is to attack with probability 40%. Note that for all possible parameters (e.g., λ's, observation capabilities, etc.), the attacker can always infer the actions of the entities and therefore obtain its reward as a function of its attacking probability and then choose the optimal one. From Fig. 3.6, it can be seen that the reward of the attacker is not necessarily positive. On the one hand, the (Bayes)-vulnerability of the attacker (i.e., $\hat{V}_X$) is measured by the probability of the entities guessing its action correctly, which satisfies $\hat{V}_X \geq 0.5$. On the other hand, due to the strategic action of the entities, their privacy loss $\sum_{j\in\mathcal{N}_c} \lambda_j \hat{V}_{Y_j}$ may be smaller than the attacker's vulnerability. That being said, when the entities perfectly know that the attacker will not attack at all, the vulnerability of the attacker is non-zero, which may result in a negative reward for the attacker. However, the reward of the outside option for the attacker (i.e., not to attack at all) could be set as 0 by designing appropriate reward functions and vulnerability functions (i.e., the gain function $g(w, x)$ in g-vulnerability) if needed.

3.4.4 Optimal Number of Collaborating Entities

In this subsection, the optimal number of collaborating entities is examined. In particular, it is assumed that all the entities have the same importance factor λ. The other parameters are the

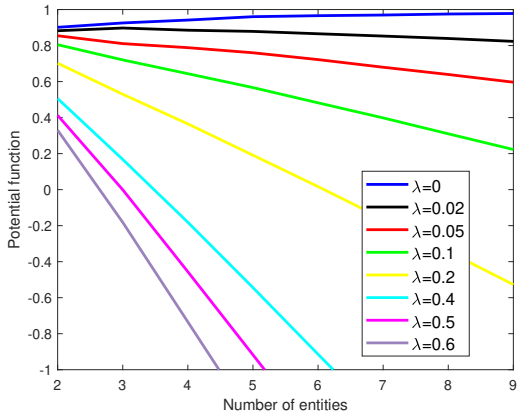

Figure 3.7: Potential function vs number of entities

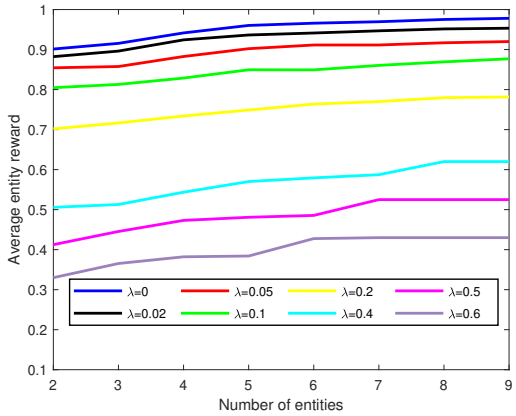

Figure 3.8: Average entity reward vs number of entities

same as those in Section 3.4.2.

Fig. 3.7 and Fig. 3.8 show the potential function (c.f. Eq. (3.38)) and the average entity reward (c.f. Eq. (3.37)) in terms of the number of entities N with different λ at NE, respectively. It can be seen that when $\lambda = 0$, the potential function (which is the opposite of the attacker's reward function) and the average entity reward increase as the number of entities increases. In this case, the entities have no concern on privacy and therefore they are willing to collaborate honestly (i.e., $p^c = 0$). As a result, as the number of entities grows, the posterior vulnerability of the attacker (and therefore the potential function and the average entity reward) increases. When λ is small, the potential function first increases as the number of entities increases and then decreases. In these cases, the collaboration gain (i.e., the increase in the posterior vulnerability of the attacker $\hat{V}_X$ by adding one entity) decreases as the number of entities grows. For the first few entities, the collaboration gains are still larger than the privacy loss (i.e., $\lambda \hat{V}_Y$) of the added entities, but they are quickly overpowered by the latter. For larger λ's, the collaboration gain can never make up for the privacy loss of any entity and the potential function decreases as the number of entities increases. However, the average entity reward behaves differently for large λ. In particular, it gradually stabilizes. The reason behind this result is that when the number of entities increases, the leader has sufficient information to estimate the action of the attacker accurately (i.e., $\hat{V}_X$ is close to 1). In this case, adding a new entity does not help much since the collaboration gain may not be able to compensate for the corresponding privacy loss and the entity will misreport with a high probability. It can be seen from Fig. 3.7 that the reward of the attacker is increasing as the number of entities grows. As a result, the optimal number of collaborating entities should be taken as the smallest N when the average entity reward achieves its stable point (e.g., $N = 7$ when $\lambda = 0.5$).

3.5 Related Works

As sophisticated large-scale attacks emerge, the performance of an individual security system is rarely satisfactory. As a result, significant research efforts have been devoted to private security-related information sharing for collaborative defense. Most of the existing works consider the scenario that the collaborative entities share their alerts or traffic data directly and prevent the information leakage from the shared information (e.g., the IP addresses) using Bloom filters [7–9], sanitization [10, 11] or anonymization and encryption methods [12–14]. [15] allows each entity to directly query the traffic data of other entities and proposes to protect the user information by ensuring differential privacy of the query results. In this chapter, a different type of privacy issue is considered. In particular, we consider the collaborative detection of the same attack, in which multiple entities share their observations (e.g., detection results) directly, and no user-specific private information is involved. Furthermore, the QIF theory is adapted to quantify the collaboration gain and privacy loss of the entities.

To guide the information sharing process for a good tradeoff between the collaboration gain and the privacy loss, various game-theoretic methods have been developed. For example, [100] proposed a two-player game model to reveal the benefit of information sharing and pointed out how the characteristics of the entities affect the incentives for information sharing. In [92] and [93], a two-player game between two competing firms that share a common platform was formulated. By game-theoretic analysis, the tradeoff between security investment and privacy breach in information sharing was studied and discussed. [94] used a two-stage Bayesian game to analyze the information sharing decisions of the two competing firms. [95] modeled the information exchange among the firms as a distributed non-cooperative game and found the best security investment and information sharing strategies. [96] considered a set of users in a public cloud who share the same hypervisors and obtained the necessary conditions under which a rational user in a public cloud will share his discovered vulnerabilities by analyzing the NE strategies of the proposed two-player game. However, the above works did not explicitly consider the optimal attacking strategy of the attacker and the privacy issues induced by information sharing. To quantitatively study the tradeoff between the utility and privacy, in the presence of an attacker, our previous works [101, 102] formulated the problem as a repeated two-layer single-leader multi-follower game and investigated the optimal collaboration strategies of the collaborative entities. The game is solved under the assumption that the attacker will stop attacking once its attack is detected by the security entities and the privacy leakage of the entities are measured by Shannon-entropy. In this chapter, we use the QIF theory to measure the vulnerability of both the attacker and the collaborative entities and solve the games without any constraint. In addition, instead of using heuristic reward functions as in our previous works, we use the quantitative information flow theory to model the collaboration gain, which captures

more general and practical settings.

There are some security games that consider the defender's information leakage induced by its actions. [103] proposed two-player zero-sum games in which a defender chose probabilities of secrets and an attacker tried to learn some of the defender's secrets. The impact of the potential information leakage on the defender's optimal strategy was investigated. [104] considered two-player games in which the attacker incrementally and stealthily obtained partial information about the defender's secret and the defender periodically changed the secret to prevent compromise of the system. [91] presented zero-sum information leakage games in which both the defender and the attacker were taking actions to minimize or maximize the information leakage of a certain secret, respectively. The utilities of both the attacker and the defender were defined as information-leakage measures. However, to the best of our knowledge, none of the existing works has explored the games that capture the secrets of both the attacker and multiple defenders while the defenders are trying to learn the attacker's secret in a collaborative and privacy-aware manner.

3.6 Conclusions

In this chapter, three games are formulated to model the interaction among collaborative security entities and the attacker, with each game corresponding to one possible scenario of interest in practice. By solving the games, the optimal strategies for the adversary and the entities are obtained. In addition, when the entities are selfish and independent, a provably convergent log-linear learning algorithm is adapted to compute the optimal collaboration strategies for the entities. Through numerical computation, we obtain the utility-privacy tradeoff curves, with which the collaborative entities can better evaluate the privacy cost of achieving certain security performance. In particular, in some scenarios, if the entities obfuscate their true observations such that their privacy loss is less than a threshold (which decreases as the number of entities increases), sharing the obfuscated observations does not help to increase the Bayes-vulnerability of the attacker. This essentially helps the entities avoid adopting ineffective collaboration strategies in practice. In addition, given the utility-privacy tradeoff curve, the optimal collaboration strategies of the entities are obtained for any possible attacking strategy. Furthermore, it is revealed that there is always an optimal attacking probability that the attacker can infer based on its observations of the entities' behavior and its knowledge about the system and the game. Finally, the simulation results show that the privacy concern changes the outcome of collaboration. More specifically, different from the scenario without privacy concern in which more collaborative entities are always preferred, there is an optimal number of participating entities when they are concerned about their privacy, which can serve as a guidance for the design of such collaborative detection groups in practice.

Chapter 4

Minimizing the Age of Information in the Presence of Location Privacy-aware Mobile Agents

In this chapter, the problem of AoI minimization in the presence of multiple location privacy-aware mobile agents in mobile crowd sensing is considered. As introduced in Chapter 1, the location differential privacy is considered as a "commodity" that the mobile agents are willing to trade with the BS. The BS first publishes its payment mechanism. After receiving the payment information, the mobile agents determine their differential privacy levels by optimizing their utility functions, which measure the difference between the expected payment they will receive and their privacy loss. In this sense, for rational mobile agents that aim to maximize their own utility, the BS can adjust its payment mechanism to motivate the mobile agents to select its desired privacy levels. In this chapter, the lower bound of the expected payment for the BS is derived as a function of the differential privacy level for each mobile agent, and the corresponding payment mechanism that achieves the lower bound is proposed. Given the privacy levels of all the mobile agents and the corresponding expected payments, the AoI minimization problem under budget constraint is formulated and a cost-efficient mobile agent selection algorithm is proposed.

The remainder of this chapter is organized as follows. Section 4.1 presents the overview of the system. The incentive mechanism is proposed in Section 4.2. Section 4.3 presents the proposed mobile agent selection algorithm. The discussion about large scale scenarios is presented in Section 4.4. The effectiveness of the proposed algorithm is demonstrated by simulations in Section 4.5. Related works are discussed in Section 4.6, and Section 4.7 concludes this chapter.

Table 4.1: Important notations

N	the number of ground terminals.
$A_n(t)$	age of information of ground terminal n at time slot t.
ϵ_i	differential privacy level of mobile agent i.
$\bar{P}_i(\epsilon_i)$	expected payment of the BS to mobile agent i.
$g_i(\epsilon_i)$	privacy loss function of mobile agent i.
C_i	information collection and delivery cost of mobile agent i.
$p^{tr}(n)$	the probability that ground terminal n determines to transmit its update.
$p(s_t)$	the probability that the set of ground terminals that determine to transmit their updates at time slot t is s_t.
l_t^i	true location of mobile agent i at time slot t.
$p(l_t^i = j)$	the probability of mobile agent i visiting ground terminal j at time slot t.
$\hat{l}_t^i$	obfuscated location of mobile agent i at time slot t.
$p_i^s(\hat{l}_t^i)$	the probability of mobile agent i being selected by the BS for information delivery if it reports an obfuscation location $\hat{l}_t^i$.
$c(t)$	budget of the BS at time slot t.
$\mathcal{S}_l$	the set of mobile agents that report the obfuscated location l and are selected by the BS for information delivery.

4.1 System Overview

4.1.1 System Model

In this chapter, a network consisting of a BS and N ground terminals (denoted by $\mathcal{N}$), which need to communicate with the BS, is considered. Important notations used in this chapter are summarized in Table 4.1. We consider a time-slotted system with slot duration normalized to unity. At each time slot, each ground terminal i independently determines whether to transmit its update or not by following some stationary policy which is controlled by or known to the BS.[1] However, the ground terminals cannot directly communicate with the BS (e.g., due to their limitations in transmit power and communication capability). Therefore, the BS outsources the information delivery tasks to some location privacy-aware mobile agents. More specifically, the BS first publishes the AoI minimization task on MCS platforms and receives necessary information from interested mobile agents (e.g., utility functions and mobility models). By accepting the information delivery tasks, the mobile agents are required to collect and deliver the updates for a pre-determined period (e.g., τ time slots) while travelling in the network.[2]

Given the set of mobile agents, an overview of the system is shown in Fig. 4.1. Specifically,

[1]For instance, it can adopt some fixed policy and transmit the updates with certain probability at each time slot. Some more complicated stationary policy (such as the one proposed in [105]) can also be considered. We note that our proposed method can deal with any possible transmission policy.

[2]Note that when $\tau = 1$, it is equivalent to the commonly considered MCS systems with one-shot tasks.

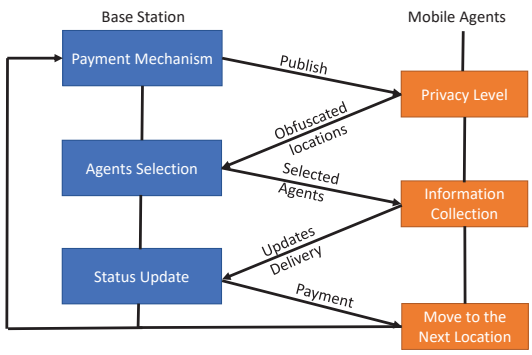

Figure 4.1: System Overview

the BS first publishes the specified non-negative payment mechanism for each mobile agent, based on which the mobile agents will determine their privacy levels and report obfuscated locations to the BS. Then, the BS selects a set of mobile agents to collect information from the terminals at their reported locations. Similarly to [23], we consider the scenario where the mobile agents can only collect information from ground terminal n when they are near the location of ground terminal n.[3] Therefore, the mobile agents that are selected for information delivery fail to collect the information if they report locations other than the true locations and the BS will know it right after. In particular, it is assumed that the mobile agents cannot send falsified information to fool the BS. For instance, the ground terminals may incorporate some encrypted information, which can be decrypted only by the BS, together with the updates. The BS can therefore verify whether the updates are from the corresponding ground terminals by checking the embedded authentication code. By reporting an obfuscated location n, the mobile agents indicate that they are in the communication range of the ground terminal n without revealing their exact location (i.e., it means that the mobile agents can be at any locations within the communication range of the ground terminal n). In addition, it is assumed that the communication ranges of the ground terminals do not overlap and the mobile agents cannot collect information from multiple ground terminals at one location (otherwise, these ground terminals can be considered as a whole and treated as one "virtual" ground terminal). After information collection, the mobile agents forward the updates to the BS and the BS will pay the mobile agents accordingly.

[3]We note that if a mobile agent is allowed to collect information from a far away ground terminal, the corresponding travel time may not be negligible, which makes the mobile agent selection highly complicated. Such a scenario will be considered in our future works.

4.1.2 Age of Information Model

In this chapter, a discrete time model is considered. In each time slot $t \in \{1, 2, 3, \cdots\}$, the BS stimulates the mobile agents to report their locations and selects a set of mobile agents to deliver the updates. AoI [22] is adopted as the key performance metric to quantify the freshness of the status information regarding the ground terminals at the BS. It is essentially defined as the elapsed time since the generation of the last status update that the BS received from the ground terminals. Let $A_n(t)$ denote the AoI of ground terminal n at the end of time slot t. Then, we have $A_n(t) = t - r(n)$, in which $r(n)$ is the time instance at which terminal n's most recent status update received by the BS is generated. If any mobile agents are selected to collect information from a ground terminal that determines to transmit an update, the ground terminal generates and sends a fresh update to the mobile agent. The same information delivery model as that in [23, 105] is adopted. More specifically, when a mobile agent collects an update from the ground terminals, the update will be relayed to the BS immediately. In addition, the update is supposed to be generated in the beginning of the time slot and delivered to the BS within one time slot without any error. In this sense, the AoI of the corresponding ground terminal drops to 1 every time an update is delivered successfully. The dynamics of $A_n(t)$ are given by

$$A_n(t) = \begin{cases} 1, & if\ status\ update\ arrives\ at\ the\ BS, \\ A_n(t-1) + 1, & otherwise. \end{cases} \tag{4.1}$$

In this sense, the instantaneous AoI of all the ground terminals is given by

$$A(t) = \frac{1}{N} \sum_{n=1}^{N} A_n(t). \tag{4.2}$$

4.1.3 Agent's Utility Model

In this chapter, M location privacy-aware mobile agents are considered. In particular, note that the mobile agents are able to receive the updates as long as they are in the communication range of the ground terminals. Therefore, in this chapter, the location privacy is considered as whether the mobile agents are in the communication range of each ground terminal. For the mobile agents that successfully deliver the information to the BS, they will receive a payment for their work, which can also be considered as a compensation for revealing their locations. However, for those mobile agents that do not deliver the information successfully (i.e., the mobile agents are not selected to perform the information collection tasks or the corresponding ground terminals decide not to transmit updates), sharing their locations induces location privacy loss without receiving any payment. Such privacy concerns in turn deter the mobile agents from

participating in the information delivery process. Therefore, the mobile agents are allowed to misreport their locations for location privacy preservation. Specifically, location differential privacy [35] is adopted to measure the privacy loss of the mobile agents caused by location sharing. Its definition is given as follows.

Definition 8. *[35] At any time slot t, a randomized mechanism $\mathcal{A}$ satisfies ϵ-differential privacy if for any output $\hat{l}_t \in \mathcal{N}$ and any two locations $x_1 \in \mathcal{N}$ and $x_2 \in \mathcal{N}$, the following holds:*

$$\frac{\Pr(\mathcal{A}(x_1) = \hat{l}_t)}{\Pr(\mathcal{A}(x_2) = \hat{l}_t)} \leq e^{\epsilon}. \tag{4.3}$$

Differential privacy ensures that the ratio of the probabilities of a mobile agent reporting any obfuscated location given any two possible true locations is no larger than e^{ϵ}, which can be controlled by choosing a suitable value for ϵ. As a result, by knowing the obfuscated location, the BS can hardly distinguish the true location of the mobile agent. According to the above definition, the privacy level ϵ is given as follows.

$$\epsilon = \max \left\{ ln \left(\frac{\Pr(\mathcal{A}(x_1) = \hat{l}_t)}{\Pr(\mathcal{A}(x_2) = \hat{l}_t)} \right), \forall \hat{l}_t, x_1 \neq x_2 \in \mathcal{N} \right\}. \tag{4.4}$$

Therefore, similar to [106, 107], the objective of each mobile agent i is to maximize its own utility given by[4]

$$U_{a,i}(\epsilon_i) = \bar{P}_i(\epsilon_i) - g_i(\epsilon_i) - \mathbb{E}[\mathbb{1}_{collection}]C_i, \tag{4.5}$$

in which ϵ_i is the differential privacy level of mobile agent i; $\bar{P}_i(\epsilon_i)$ is the expected payment of the BS; $g_i(\epsilon_i)$ is the privacy loss function of mobile agent i, assumed to be continuously differentiable; $\mathbb{1}_{collection}$ is an indicator function for the event that mobile agent i collects and delivers the information successfully and the expectation is over the probability of successfully information delivery; C_i is the corresponding information collection and delivery cost. It is assumed that the participating mobile agents are willing to trade their location privacy for payment and $g_i(\epsilon_i)$ can be set properly to capture their privacy requirements.[5]

4.1.4 Agent's Mobility Model

In this chapter, it is assumed that the ground terminals form a graph and mobile agents are traveling inside the network along the connecting edge by following the Graph Random Waypoint

[4]We note that the mobile agent may use any randomized mechanisms to achieve ϵ-differential privacy. In the context of differential privacy, two mechanisms with the same ϵ lead to the same privacy loss. Therefore, the notation of the randomized mechanism is omitted for the ease of presentation.

[5]For the mobile agents that are only willing to trade their location privacy up to a certain level ϵ_{max}, they can also share such information with the BS such that the BS will not select an $\epsilon > \epsilon_{max}$.

(GRW) mobility model [108]. In particular, the probability of a mobile agent selecting its next location is captured by the following mobility matrix.[6]

$$\mathcal{M} = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,N} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ p_{N,1} & p_{N,2} & \cdots & p_{N,N} \end{bmatrix}$$

In particular, $p_{n,j}$ is the probability that the mobile agent chooses to move to the location of ground terminal j in the next time slot given its current location of ground terminal n.[7] Note that in practice, the mobile agent may not be able to move from location n to location j due to geographic constraints, which can be captured by setting $p_{n,j} = 0$. It is assumed that the mobility models of the mobile agents are known to the BS. We note that the main goal of this chapter is to protect the current true locations of mobile agents. In this sense, compared to the true locations, the mobility models are of less privacy concern. In addition, the BS can also obtain the probability of the mobile agents being at different locations from observations or past experiences [34]. The impact of the knowledge about the mobile agents' mobility models will be left as our future work.

Given the history of the reported locations of a mobile agent and the fact whether it delivers the information successfully, the BS is able to update and store the probability of the mobile agent visiting each ground terminal. In particular, if a mobile agent delivers the information for ground terminal n successfully at time slot t, the probability of this mobile agent being at location j at time slot $t+1$ is $p_{n,j}$. Let l_t and $\hat{l}_t$ denote the true location and the obfuscated location of the mobile agent, respectively. If the mobile agent reports an obfuscated location n but is not selected to collect information, the probability of this mobile agent being at location j at time slot $t+1$ is given by

$$\begin{aligned} \Pr(l_{t+1} = j | \hat{l}_t = n) &= \sum_{k=1}^{N} \Pr(l_{t+1} = j | l_t = k) \Pr(l_t = k | \hat{l}_t = n) \\ &= \sum_{k=1}^{N} \Pr(l_{t+1} = j | l_t = k) \frac{\Pr(\hat{l}_t = n | l_t = k) \Pr(l_t = k)}{\sum_{m=1}^{N} \Pr(\hat{l}_t = n | l_t = m) \Pr(l_t = m)}, \end{aligned} \tag{4.6}$$

in which $\Pr(l_{t+1} = j | l_t = k) = p_{k,j}$ is the probability that the mobile agent moves from location k to j, which is determined by its mobility matrix; $\Pr(\hat{l}_t = n | l_t = k)$ is the probability that the mobile agent reports location n given its true location k, which is determined by its location obfuscation strategy that will be discussed in Section 4.2; $\Pr(l_t = k)$ is the probability

[6] Note that the mobility models of different mobile agents can be different.

[7] For the ease of presentation, we denote location n as the location of ground terminal n in the following discussion.

distribution of the mobile agent's location at time slot t from the BS's perspective, which is locally stored and updated by the BS.

Finally, if the mobile agent, which reports an obfuscated location n and is selected to collect information, fails to deliver the information, the probability of this mobile agent being at location j at time slot $t+1$ is given by

$$\begin{aligned}\Pr(l_{t+1}=j|\hat{l}_t=n,fail) &= \sum_{k=1}^{N}\Pr(l_{t+1}=j|l_t=k)\Pr(l_t=k|\hat{l}_t=n,fail)\\ &= \sum_{k=1}^{N}\Pr(l_{t+1}=j|l_t=k)\frac{\Pr(\hat{l}_t=n,fail|l_t=k)\Pr(l_t=k)}{\sum_{m=1}^{N}\Pr(\hat{l}_t=n,fail|l_t=m)\Pr(l_t=m)},\end{aligned} \tag{4.7}$$

in which $\Pr(\hat{l}_t=n,fail|l_t=k)$ is the probability of the mobile agent reporting the location n but failing to deliver the information, given its true location k. Particularly, we have

$$\Pr(\hat{l}_t=n,fail|l_t=k)=\begin{cases}\Pr(\hat{l}_t=n|l_t=n)(1-p^{tr}(n)),\ if\ k=n,\\ \Pr(\hat{l}_t=n|l_t=k),\quad otherwise,\end{cases} \tag{4.8}$$

in which $p^{tr}(n)$ is the probability that ground terminal n determines to transmit its update. In addition, if a new mobile agent joins the information delivery process, the probability of it visiting each ground terminal depends on the prior knowledge of the BS (uniform distribution if no prior knowledge is available).

4.2 The Proposed Incentive mechanism

4.2.1 Payment Mechanism of the BS

In practice, the objective of the BS is to minimize the expected AoI given a limited budget. Intuitively, in order to achieve a smaller expected AoI, the BS should stimulate the mobile agents to trade more location privacy (i.e., larger ϵ_i's), which requires a larger payment. Therefore, a smaller payment for fixed ϵ_i's allows the BS to select larger ϵ_i's given a budget, which leads to better AoI performance. With such consideration, the objective of the payment mechanism is to minimize the payment for any fixed ϵ_i's. Let l_t^i and $\hat{l}_t^i$ denote the true location and the obfuscated location of mobile agent i, respectively. In this section, it is first assumed that the obfuscation strategy of each mobile agent i is given by

$$\Pr(\hat{l}_t^i=j|l_t^i=j)=\frac{e^{\epsilon_i}}{N-1+e^{\epsilon_i}}, \tag{4.9}$$

and

$$\Pr(\hat{l}_t^i = k | l_t^i = j) = \frac{1}{N - 1 + e^{\epsilon_i}}, \forall k \neq j \in \mathcal{N}. \tag{4.10}$$

It can be verified that the above obfuscation strategy satisfies ϵ_i-differential privacy and is commonly used in the randomized response mechanism to ensure differential privacy (see, e.g., [109]).

Let $p_i^s(\hat{l}_t^i) > 0$ denote the probability of the mobile agent i being selected by the BS for information delivery if it reports an obfuscation location $\hat{l}_t^i$. Given the above obfuscation strategy, the following theorem can be proved.

Theorem 10. *Suppose that all the mobile agents intend to maximize their own payoffs. To incentivize a mobile agent i to report its location with ϵ_i-differential privacy, the expected lower bound of the payment $\bar{P}_i(\epsilon_i)$ from the BS in the multi-agent scenario is given by*

$$\begin{aligned} \bar{P}_i(\epsilon_i) \geq & [\frac{1}{N-1} - \frac{1}{N - 1 + e^{\epsilon_i}}] \frac{g_i'(\epsilon_i)(N - 1 + e^{\epsilon_i})^2}{e^{\epsilon_i}} \\ & + \sum_{s_t} p(s_t) \sum_{j=1}^{N} p(l_t^i = j) \frac{e^{\epsilon_i}}{N - 1 + e^{\epsilon_i}} p_i^s(j) \mathbb{1}(j \in s_t) C_i, \end{aligned} \tag{4.11}$$

in which $p(s_t)$ is the probability that the set of ground terminals that determine to transmit their updates at time slot t is s_t. $g_i(\cdot)$, l_t^i and C_i are the privacy loss function, the true location and the information collection cost of mobile agent i, respectively. $p_i^s(\cdot)$ is the probability of mobile agent i being selected.

Proof. Please see Appendix C.1. □

Remark 9. *In the considered application, the expected AoI depends only on whether a mobile agent is in the communication range of the ground terminals. That being said, reporting any location inside the communication range of a ground terminal results in the same utility for the BS and therefore discrete obfuscation mechanism is used, among which the randomized response mechanism and the exponential mechanism [40] are most commonly used. We focus on the randomized response mechanism given in (4.9) and (4.10) in this chapter. In addition, (4.11) is general in terms of the privacy loss function $g_i(\epsilon)$ and it holds for any continuous and differentiable $g_i(\epsilon)$.*

It can be learned from Theorem 10 that given the randomized response mechanism, the BS cannot pay less than the lower bound in (4.11) to stimulate mobile agent i to obfuscate its true location with ϵ_i-differential privacy. This essentially allows the BS to evaluate its payment mechanism. In particular, any payment mechanism that achieves the lower bound is optimal for the BS. In addition, it also specifies the BS's cost for certain privacy level ϵ_i that further determines the expected AoI.

Remark 10. *Note that the mobile agent selection probability $p_i^s(\hat{l}_t^i)$ depends on the BS's mobile agent selection strategy, which may consider lots of factors in practice (e.g., AoI of the network, the BS's budget, and the reported locations of the other mobile agents). For the ease of presentation, we use the notation $p_i^s(\hat{l}_t^i)$ since only $\hat{l}_t^i$ can be controlled by each individual mobile agent i. In particular, in the multi-agent scenarios, $p_i^s(\hat{l}_t^i)$ can be evaluated by $p_i^s(\hat{l}_t^i) = \mathbb{E}[\mathbb{1}(\hat{l}_t^i)]$, where $\mathbb{1}(\hat{l}_t^i)$ is an indicator function for the event that the mobile agent i is selected given its reported location $\hat{l}_t^i$ and the mobile agent selection strategy, and the expectation is over the reported locations of all the other mobile agents.*

Note that since mobile agent i's selection of optimal differential privacy level ϵ_i depends on both the payment mechanism and its privacy loss function $g_i(\epsilon_i)$, it is difficult, if not impossible, to design a payment mechanism without knowing $g_i(\epsilon_i)$ (and therefore the utility function of mobile agent i). To facilitate the discussion, we consider the following privacy loss function for the mobile agent.

$$g_i(\epsilon_i) = L_i(e^{\epsilon_i} - 1), \tag{4.12}$$

in which L_i is some positive constant. Some explanations about (4.12) are in order. On the one hand, according to [35], if mobile agent i adopts ϵ_i-differential privacy strategy, the knowledge about mobile agent i's true location that an adversary, with any prior knowledge, can obtain is bounded by $\frac{p(l_t^i=j|\hat{l}_t^i)}{p(l_t^i=j)} \leq e^{\epsilon_i}, \forall j$, in which $p(l_t^i = j)$ and $p(l_t^i = j|\hat{l}_t^i)$ are the prior and posterior probability distribution of mobile agent's location from the adversary's view, respectively. In this sense, (4.12) conforms to this intuition that the privacy loss grows exponentially as ϵ_i increases. On the other hand, it can be observed that when $\epsilon_i = 0$, i.e., the most strict requirement where sharing the obfuscated location does not leak any information, the privacy loss satisfies $g_i(0) = 0$.

Given the above privacy loss function, the following incentive mechanism, which can achieve the lower bound, is proposed. In particular, since the incentive mechanism is the same for all the mobile agents, the indices of the mobile agents are ignored in the following discussion for the ease of presentation.

The Proposed Incentive Mechanism:

- In the beginning of each time slot t, depending on the corresponding level of privacy ϵ (which can be different for different mobile agents) that it expects the mobile agent to select, the BS publishes the mobile agent selection probability $p^s(\hat{l}_t), \forall \hat{l}_t \in \mathcal{N}$ and the

following payment mechanism.[8]

$$P(\epsilon) = \begin{cases} \frac{g'(\epsilon)(N-1+e^{\epsilon})^2}{p^s(\hat{l}_t)p^{tr}(\hat{l}_t)(N-1)e^{\epsilon}} + C, \ \textit{if } \hat{l}_t = l_t \textit{ and } \hat{l}_t \in s_t, \\ 0, \qquad\qquad\qquad\qquad \textit{Otherwise}, \end{cases} \tag{4.13}$$

in which $p^{tr}(\hat{l}_t) = \sum_{s_t} p(s_t)\mathbb{1}(\hat{l}_t \in s_t)$ is the probability that ground terminal $\hat{l}_t$ determines to transmit its update.

- Then based on its true location l_t and the payment mechanism, the mobile agent selects a privacy level that maximizes its own utility and sends an obfuscated location $\hat{l}_t$ to the BS. The BS then determines whether the mobile agent is selected to deliver the updates according to the mobile agent selection strategy, which is subject to design.
- If the mobile agent is selected for information delivery and $\hat{l}_t \in s_t$, i.e., ground terminal $\hat{l}_t$ determines to transmit its update, it collects the update from ground terminal $\hat{l}_t$ and delivers it to the BS (successfully only when $l_t = \hat{l}_t$).
- The BS verifies whether $\hat{l}_t = l_t$ and $\hat{l}_t \in s_t$ and determines the final payment according to (4.13).

Remark 11. *The criteria $\hat{l}_t = l_t$ and $\hat{l}_t \in s_t$ are easy to verify. Since the mobile agent can only collect the information successfully from the ground terminals that are at its reported location and determine to transmit their updates, the BS can determine whether $\hat{l}_t = l_t \in s_t$ or not by checking if the mobile agent delivers the information successfully. In addition, it can be verified that $U_a(\epsilon) = \frac{L(N-1+e^{2\epsilon})}{N-1} > 0$ given the above incentive mechanism.*

In general, when the budget is limited, the probability of a mobile agent being selected $p^s(\cdot)$ decreases as the number of mobile agents increases. According to (4.13), to stimulate a mobile agent to report with the same differential privacy level, the payment required for each successful information delivery increases. Intuitively, as the probability of being selected decreases, the utility of a mobile agent decreases since there is always a location privacy loss regardless of whether it is selected or not. Therefore, a larger payment for successful information delivery is needed. Such a phenomenon is important but largely ignored in the existing works (e.g., [40]).

Theorem 11. *The proposed incentive mechanism achieves the lower bound in Theorem 10 for any privacy loss function $g(\epsilon)$ that satisfies*

$$\frac{g'(x)}{g'(y)} \leq \frac{e^x}{e^y}, \forall 0 \leq x \leq y. \tag{4.14}$$

[8]Note that (4.13) is constructed according to the condition that the equality in (4.11) holds, which is given in Appendix C.1. In addition, we note that ϵ is determined by the BS and the payment mechanism is designed such that the optimal privacy level for the mobile agent is also ϵ.

Proof. Please see Appendix C.2. □

Remark 12. *It can be verified that the privacy loss function in (4.12) satisfies (4.14). The optimal incentive mechanisms for other privacy loss functions are left as future work.*

To this end, some explanations about the proposed incentive mechanism are in order. Firstly, Theorem 10 shows that the lower bound of the payment to mobile agent i depends on the privacy level ϵ_i and the probability of mobile agent i being selected $p_i^s(\cdot)$. To minimize the AoI given a budget $c_i(t)$ for mobile agent i, it is expected that $\bar{P}_i(\epsilon_i) = c_i(t)$. As a result, the corresponding lower bound and upper bound of ϵ_i can be obtained by setting $p_i^s(\cdot) = 1$ and $p_i^s(\cdot) = 0$ in the right-hand side of (4.11), respectively.

Similarly, in the multiple mobile agent scenario, given the total budget $\sum_{i=1}^{M} \bar{P}_i(\epsilon_i) = c(t)$, the feasible region of ϵ_i's can be obtained. Secondly, given the payment mechanism, Theorem 11 guarantees that the privacy levels selected by the mobile agents are the same as the ϵ_i's determined by the BS, and the expected payment is minimized and given by the right-hand side of (4.11). Finally, the probability of each mobile agent being selected depends on the BS's mobile agent selection strategy. From the above discussion, it can be seen that to minimize the AoI given a budget constraint, the BS has to jointly optimize the privacy levels ϵ_i's and the mobile agent selection strategy. For instance, the BS may assign a larger ϵ to the mobile agents (corresponding to higher payments) that are more likely to visit the ground terminals with higher AoI, and select them with a higher probability, which leads to better AoI performance. In this chapter, we mainly investigate the mobile agent selection strategy with fixed ϵ_i and budget constraint, which is deferred to Section 4.3. The optimization of ϵ_i's is highly non-trivial and left as future work. However, our payment mechanism is general and can be applied together with any realization of ϵ_i's and mobile agent selection strategies.

4.2.2 Optimality of the Obfuscation Strategy

In the above discussion, it is assumed that the obfuscation strategy is given by (4.9) and (4.10). Next, we show the optimality of such a symmetric obfuscation strategy based on the following assumptions.

Assumption 2. *Given the mobile agent's true location i, it reports its location truthfully with probability $p_i \geq \frac{1}{N}$ and a misreported location j with probability $\frac{1-p_i}{N-1}, \forall j \neq i$.*

Note that the first condition in Assumption 2 (i.e., $p_i \geq \frac{1}{N}$) is commonly used in the randomized mechanisms for differential privacy (e.g., [109]). The second condition indicates that when the mobile agent determines to misreport its location, it will randomly select the other locations with the same probability. Considering that the mobile agent will not receive any

payment when it misreports its location, such an assumption is reasonable. Given Assumption 2, by plugging $p_i, \forall i$ into (4.4), the privacy level ϵ is reduced to

$$\epsilon = \ln\Big(\max\Big\{\frac{(N-1)p_i}{1-p_j}, \frac{1-p_j}{(N-1)p_i}, \frac{1-p_i}{1-p_j}, \frac{1-p_j}{1-p_i}, \forall i, j\Big\}\Big). \tag{4.15}$$

Assumption 3. *The probability of the mobile agent visiting any location is no larger than* $\frac{1}{2}$*, i.e.,* $p(l_t = i) \leq \sum_{j \neq i} p(l_t = j), \forall i \in \mathcal{N}$*.*

Remark 13. *We note that the above assumption conforms to the intuition that in order to minimize the AoI as well as preserving location privacy, it is better for the mobile agent to visit each ground terminal more evenly.*

Theorem 12. *For the payment mechanism proposed above, given Assumption 2 and Assumption 3, the obfuscation strategy given by (4.9) and (4.10) is optimal for the mobile agent.*

Proof. Please see Appendix C.3. □

4.3 Mobile Agent Selection

In this section, the mobile agent selection strategy is discussed. In particular, it is assumed that there are always more than one mobile agent available for selection. Note that the payment in (4.11) and the AoI depend on the mobile agent selection strategy (i.e., $p_i^s(\cdot)$). In addition, considering that the budget of the BS is usually limited in practice, an effective mobile agent selection strategy is in need. In particular, given the AoI of the ground terminals at time t (i.e., $A_i(t), \forall i \in \mathcal{N}$) and the desired differential privacy level of each mobile agent $\epsilon_j, \forall 1 \leq j \leq M$, the objective is to minimize the expected AoI $\mathbb{E}[A(t+1)]$ given the budget $c(t)$. The corresponding optimization problem can be formulated as follows.[9]

$$\begin{aligned} &\min_{\mathcal{S}} \mathbb{E}[A(t+1)] \\ \text{s.t.} \quad &\sum_{i \in \mathcal{S}} \bar{P}_i(\epsilon_i) \leq c(t), \end{aligned} \tag{4.16}$$

[9]Note that it may be desirable to minimize the time-average AoI $\frac{1}{T}\sum_{t=1}^{T} A(t)$, which is determined by the budget allocated to each time slot $c(t)$, the privacy levels of the mobile agents ϵ_i's and the mobile agent selection strategy. However, joint optimization over these parameters is highly non-trivial and left as our future work. In this chapter, the mobile agent selection problem given $c(t)$ and ϵ_i's is considered. In such a case, minimizing the expected time average AoI is equivalent to minimizing $\mathbb{E}[A(t+1)]$.

in which $\mathcal{S}$ is the set of mobile agents to be selected by the BS. In particular, the expected AoI $\mathbb{E}[A(t+1)]$ is given by

$$\mathbb{E}[A(t+1)] = \frac{1}{N}\sum_{k=1}^{N}\left[(1-p^{tr}(k))[A_k(t)+1] + p^{tr}(k)\Big[[\prod_{i\in\mathcal{S}}(1-q_i^k)]A_k(t)+1\Big]\right], \tag{4.17}$$

in which $p^{tr}(k)$ is the probability that ground terminal k determines to transmit its update; q_i^k is the probability of mobile agent i delivering the update from ground terminal k successfully; $A_k(t)$ is the AoI of ground terminal k at time slot t. In particular, we have

$$q_i^k = p(l_t^i = k)\frac{e^{\epsilon_i}}{N-1+e^{\epsilon_i}}p_i^s(\hat{l}_t^i = k), \tag{4.18}$$

in which $l_t^i = k$ means that the true location of mobile agent i at time t is k and $p(l_t^i = k)$ is the corresponding probability; $p_i^s(\hat{l}_t^i = k)$ is the probability of mobile agent i being selecting when its reported location is $\hat{l}_t^i = k$.

In addition, similar privacy loss functions as that in (4.12) are considered for the mobile agents, i.e.,

$$g_i(\epsilon_i) = L_i(e^{\epsilon_i} - 1), \tag{4.19}$$

in which L_i is some positive constant for mobile agent i.

Note that the optimization problem (4.16) can be considered as an integer programming problem with constraints, and therefore is difficult to solve in general. As a result, a sub-optimal greedy algorithm is proposed in this section. Specifically, before introducing the algorithm, we have the following proposition.

Proposition 3. *Let $\mathcal{S}_l$ denote the set of selected mobile agents that report an obfuscated location $l \in \mathcal{N}$, with the probabilities of successful information delivery being $q_i^l, \forall i \in \mathcal{S}_l$. Considering a sequence of virtual expected AoI $\hat{A}_l^j(t), 1 \le j \le |\mathcal{S}_l|$ and $\hat{A}_l^0(t) = A_l(t)$ in which $|\mathcal{S}_l|$ is the size of $\mathcal{S}_l$ and $\hat{A}_l^j(t) = (1-q_j^l)\hat{A}_l^{j-1}(t)$, then*

$$\mathbb{E}[A_l(t+1)] = (1-p^{tr}(l))A_l(t) + p^{tr}(l)\hat{A}_l^{|\mathcal{S}_l|}(t) + 1, \tag{4.20}$$

in which the expectation is over the probabilities of successful information delivery.

Remark 14. *Note that instead of determining the set of selected mobile agents simultaneously, Proposition 3 essentially allows us to evaluate the expected AoI reduction of selecting each individual mobile agent by introducing the virtual expected AoI sequence. In particular, the first and third term in (4.20) are independent of $\mathcal{S}_l$. As a result, the difference between $p^{tr}(l)\hat{A}_l^{j-1}(t)$ and $p^{tr}(l)\hat{A}_l^j(t)$ in Proposition 3 can be understood as the expected AoI reduction by selecting j-th mobile agent in $\mathcal{S}_l$.*

Algorithm 7 Mobile Agent Selection Algorithm

1: Initialization: The AoI of each ground terminal: $A_i(t), \forall i \in \mathcal{N}$; the average budget at time t: $c(t)$; the reported location of each mobile agent j: $\hat{l}_t^j = k_j$; the set of selected mobile agents: $S = \emptyset$; Virtual AoI sequence $\hat{A}_i^{z_i}(t) = A_i(t)$, with $z_i = 0, \forall i \in \mathcal{N}$.

2: Compute the budget for information collection costs: $\hat{c}(t) = c(t) - \sum_{j=1}^{M} [\frac{1}{N-1} - \frac{1}{N-1+e^{\epsilon_j}}] \frac{g_j'(\epsilon_j)(N-1+\epsilon_j)^2}{e^{\epsilon_j}}$.

3: **while** $\hat{c}(t) \geq \min_j p(l_t^j = k_j | \hat{l}_t^j = k_j) p^{tr}(k_j) C_j$ **do**

4: For each mobile agent j, compute the expected AoI reduction $\hat{A}_{k_j}^{z_{k_j}}(t) p(l_t^j = k_j | \hat{l}_t^j = k_j) p^{tr}(k_j)$ and the expected payment (for information collection cost) $p(l_t^j = k_j | \hat{l}_t^j = k_j) p^{tr}(k_j) C_j$, then find (with random tie breaking)

$$j = \arg\max_{j \notin S} \Bigg\{ \frac{\hat{A}_{k_j}^{z_{k_j}}(t)}{C_j} | p(l_t^j = k_j | \hat{l}_t^j = k_j) p^{tr}(k_j) > 0, \\ p(l_t^j = k_j | \hat{l}_t^j = k_j) p^{tr}(k_j) C_j \leq \hat{c}(t) \Bigg\}. \tag{4.21}$$

5: Add j into set S: $S = S \cup j$.

6: Update the virtual AoI sequence by

$$\hat{A}_{k_j}^{z_{k_j}+1}(t) = (1 - p(l_t^j = k_j | \hat{l}_t^j = k_j)) A_{k_j}^{z_{k_j}}(t), \tag{4.22}$$

$$z_{k_j} = z_{k_j} + 1. \tag{4.23}$$

7: Update the budget for information collection costs by

$$\hat{c}(t) = \hat{c}(t) - p(l_t^j = k_j | \hat{l}_t^j = k_j) p^{tr}(k_j) C_j. \tag{4.24}$$

8: **end while**

9: Return S.

Based on the idea in Proposition 3, the proposed mobile agent selection strategy is given in Algorithm 7. In particular, note that the first term in the payment (4.11) does not depend on the mobile agent selection strategy, which can be considered as the cost to stimulate each mobile agent to report its location with the BS's desired differential privacy level. Therefore, we first compute the budget for information collection costs in step 2. Then, based on the results in Proposition 3, the (virtual) AoI reduction per unit cost for each mobile agent is evaluated and the mobile agents are greedily selected in step 4-5. Finally, the virtual AoI sequence and the remaining budget are updated in step 6-7.

Algorithm 8 Large-scale Information Collection and Delivery Process

Initialization: The AoI of each ground terminal: $A_i(0), \forall i \in \mathcal{N}$; the total number of time slots: T; the average budget per time slot: c^M; the total payment: $P = 0$.
for $t = 0, 1, 2, \cdots, T$ **do**
    Update the average budget allocated to each sub-network $\mathcal{N}^b$ at time t: $c_b(t) = \frac{c^M T - P}{T-t+1} \frac{\sum_{i \in \mathcal{N}^b} A_i(t)}{\sum_{i \in \mathcal{N}} A_i(t)}, \forall 1 \leq b \leq B$.
    The BS publish the incentive mechanism described in Section 4.2.
    The mobile agents report their obfuscated locations.
    The BS Selects the set of mobile agents in each sub-network for information delivery by running Algorithm 7.
    The selected mobile agents try to collect and deliver the information, the AoI of the network is updated according to (4.1).
    The BS updates the total payment P.
end for

4.4 Large-scale Scenarios

In practice, for large-scale MCS scenarios with a large number of ground terminals, it is usually not optimal for the mobile agents to select their obfuscated locations from $\mathcal{N}$. For instance, from the BS's perspective, the probability of a mobile agent visiting somewhere far away from her working/living area is low. Therefore, the BS would prefer not to select the mobile agents reporting the locations that they rarely visit due to the low probability of successful information delivery. Nonetheless, the mobile agents still suffer from the location privacy loss for sharing the obfuscated locations. One simple but effective solution is to divide the whole network into multiple sub-networks according to the mobility models of the mobile agents such that they can select their obfuscated locations from the sub-networks. For the ease of discussion, we assume that the network is divided into B sub-networks and each mobile agent will travel only inside one corresponding sub-network. With such consideration, the information collection and delivery process in the large-scale scenarios is summarized in Algorithm 8.

Remark 15. *In Algorithm 8, we assume that there is a total budget $c^M T$ and the remaining budget is evenly allocated to the future time slots. At each time slot, the budget allocated to each sub-network is proportional to its AoI. In addition, the differential privacy levels of the mobile agents (and therefore the budget allocated to each mobile agent) are assumed to be pre-determined. However, our proposed incentive mechanism and mobile agent selection algorithm can deal with any budget allocation strategy over the time, the sub-networks as well as the mobile agents. The goal of this chapter is to introduce the proposed incentive mechanism with the desired properties, and the optimization of the budget allocation strategy will be considered in our future work.*

4.5 Simulation Results

In this section, the performance of the proposed incentive mechanism and mobile selection algorithm is examined by simulation. It is assumed that the network is divided into B sub-networks, each containing 20 ground terminals that adopt fixed stationary policy and transmit their updates with a probability of 0.5. In particular, we consider a 4×5 grid for each sub-network and the mobile agents can move from one ground terminal to another if they are connected.[10] The information delivery tasks consist of $T = 3000$ time slots. In addition, two types of mobile agents are considered. Type-1 mobile agents will stay at their current location i with a probability of p_i and move to each neighboring ground terminal in the corresponding sub-network b with the same probability of $\frac{1-p_i}{|\mathcal{N}_i^b|}$, in which $\mathcal{N}_i^b$ is the set of neighboring ground terminals in sub-network b; Type-2 mobile agents are more AoI sensitive, in that they will visit the neighboring ground terminals with probabilities that are proportional to their AoI. In particular, the probability of a type-2 mobile agent at ground terminal k visiting a neighboring ground terminal i is given by $\frac{A_i(t)}{\sum_{j \in \mathcal{N}_k^b} A_j(t)}$.[11]

4.5.1 The Performance of the Proposed Incentive Mechanism

In this subsection, one sub-network is considered and the performance of the proposed payment mechanism is examined. First of all, we examine the payment required to stimulate the mobile agents to obfuscate their locations with desired differential privacy levels (i.e., ϵ). In particular, it is assumed that the BS's budget is large enough such that all the mobile agents will be selected. To the best of our knowledge, there is no existing payment mechanisms in the literature that stimulate the mobile agents to select the BS's desired differential privacy levels. Therefore, we compare the proposed incentive mechanism with some naive mechanisms. A natural idea is to pay the mobile agents more when they reports their true locations and less otherwise. Since the true locations of the mobile agents are unknown, we assume that the BS utilizes its prior knowledge and considers the locations with the highest probabilities as the mobile agents' "true" locations in the naive counterparts. In addition, the BS pays the mobile agents whenever they report the "true" locations and the payments are determined analytically (similarly to the procedures in the proof of Theorem 11) such that the utilities of the mobile agents are maximized when they select the corresponding ϵ. For the "Lower Bound" counterpart, we compute the payment in (4.11). Fig. 4.2 shows the required payments to stimulate a type-2 mobile agent with the privacy loss factor $L = 1$ and the information collection cost $C = 0$ to select a differential

[10] In the simulations, it is assumed that the mobile agents only travel inside the sub-network that they belong to. However, it can be easily generalized to the scenario that they can travel across the sub-networks by modifying their mobility models.

[11] Note that our proposed method can also be applied in other graphs, mobility models and system parameters, and qualitatively similar simulation results can be obtained, which is omitted in the interest of space.

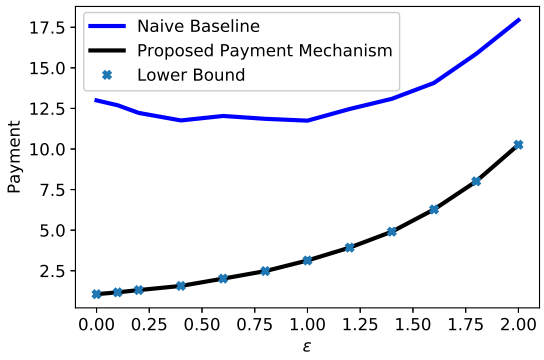

Figure 4.2: $\bar{P}(\epsilon)$ vs ϵ

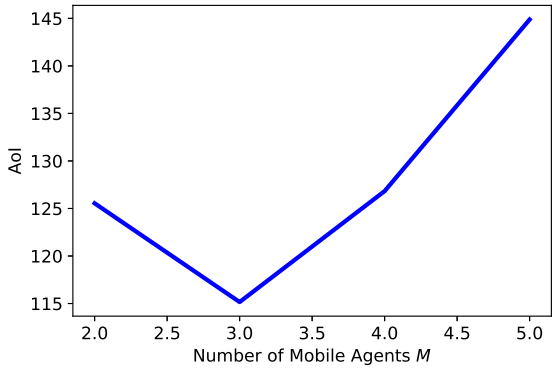

Figure 4.3: AoI vs the Number of Mobile Agents

privacy level of ϵ. We note that for the baseline, the payment also depends on the BS's prior knowledge about the mobile agent's true location and a smaller payment is required when the BS can estimate the mobile agent's location more accurately. A larger ϵ (i.e., the mobile agent reports its true location with a higher probability) in general indicates better prior knowledge in the following time slots and therefore smaller payments. On the other hand, the instant payment at each time slot is an increasing function of ϵ. As a result, as ϵ increases, the payment in the baseline first decreases and then increases. It can be observed that compared to the baseline, the proposed mechanism reduces the payment by over 90%, 70% and 40% when ϵ is 0, 1 and 2, respectively, which demonstrates its effectiveness. In addition, it can be observed that the payment of the proposed mechanism is essentially the same as that of the lower bound, which verifies Theorem 11.

4.5.2 The Impact of the Number of Mobile Agents

In this subsection, one sub-network is considered and the impact of the number of mobile agents M is examined. In particular, when $M = 2$, there is one type-1 mobile agent and one type-2 mobile agent. Each time we increase M, one type-2 mobile agent and one type-1 mobile agent are added in turn. The privacy loss factors L's of all the mobile agents are set as 0.3 and the information collection costs C's of the type-1 mobile agents and the type-2 mobile agents are set as 40 and 20, respectively. In addition, the staying probabilities of the type-1 mobile agents are set as $p_i = 0.5, \forall i \in \mathcal{N}$.

Fig. 4.3 compares the AoI for different number of mobile agents when the proposed Algorithm 7 is implemented, with the privacy levels of all the mobile agents being $\epsilon = 1$. When $M = 2$, both mobile agents are always selected and the corresponding average payment is obtained. When $M > 2$, we set the BS's average budget the same as that in the $M = 2$ case. It can

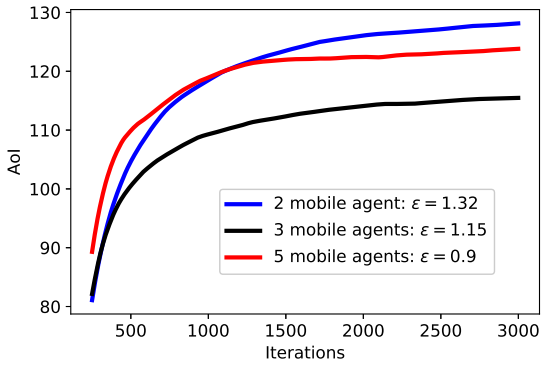

Figure 4.4: AoI vs the Number of Mobile Agents

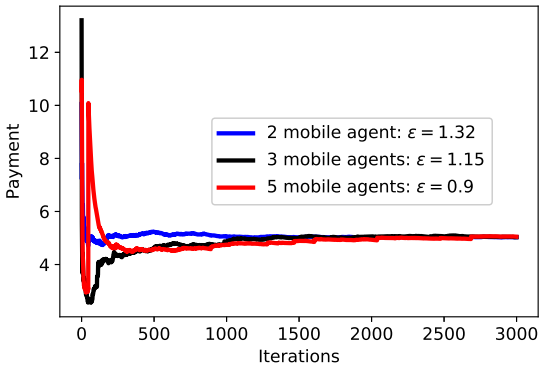

Figure 4.5: Payment vs the Number of Mobile Agents

be observed that as the number of mobile agents increases, the AoI first decreases and then increases. This is because when the number of mobile agents is small, increasing the number of mobile agents helps since more mobile agents are available for selection. When the number of mobile agents is large enough, the probability of being selected for the mobile agents decreases as the number of mobile agents increases. Therefore, a larger payment is required to stimulate each mobile agent to select the desired privacy level. As a result, fewer mobile agents can be selected, which results in larger AoI.

In addition, we also examine the scenario in which the BS selects a fixed number of mobile agents at each iteration. In particular, at each iteration, only one mobile agent is selected according to step 4 in Algorithm 7 (i.e., set $c(t)$ as infinity and stop running Algorithm 7 when $|S| = 1$). The privacy levels ϵ of all the mobile agents are the same and selected such that the average payment of the BS is 5. Fig. 4.4 and Fig. 4.5 show the AoI performance and the BS's average payment. It can be observed that, as the number of mobile agents increases, the AoI also first decreases and then increases. When the number of mobile agents is large enough, the probability of being selected for the mobile agents decreases as the number of mobile agent increases. As a result, for the same payment, the mobile agents will select smaller ϵ's, which leads to larger AoI.

From the above results, it can be learned that allocating the budget to more mobile agents may lead to worse AoI performance. It may be better for the BS to restrict the number of mobile agents in each sub-network (e.g., by offering its payment mechanism to a small subset of the available mobile agents). Such insights are not captured in the existing methods since they consider fixed privacy levels (i.e., the mobile agents do not adjust their privacy levels based on the payment mechanism).

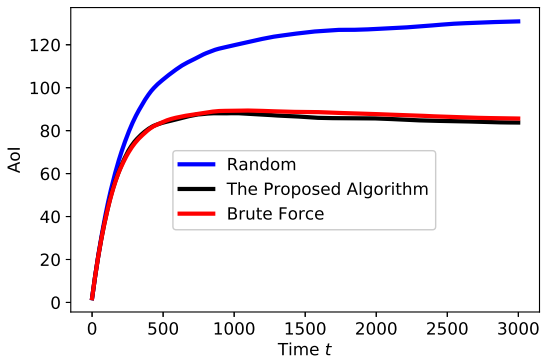

Figure 4.6: AoI of Algorithm 7

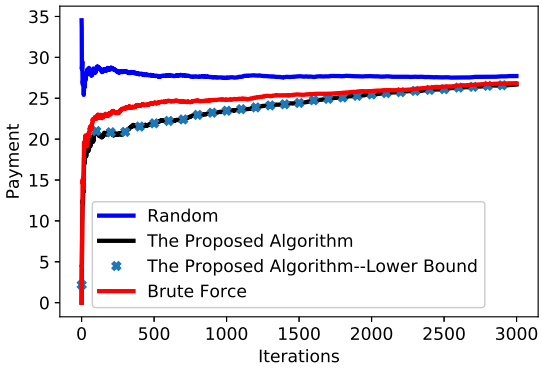

Figure 4.7: Average Payment of Algorithm 7

4.5.3 The Effectiveness of Algorithm 7

In this subsection, the performance of the proposed mobile agent selection algorithm is examined. In particular, we consider the scenario in which there are two type-2 mobile agents with privacy loss factor $L_1 = 2$ and $L_2 = 1$, respectively, and one type-1 mobile agent with $L_3 = 5$ in each sub-network. In particular, the type-1 mobile agent stays at their current locations with probabilities $p_i = 0.5, \forall i \in \mathcal{N}$. The information collection cost of the mobile agents are set as 10, 20 and 40, respectively. The privacy levels of all the mobile agents are set as $\epsilon = 1$. The performance of Algorithm 7 is compared with two baselines. In the "Random" baseline, two mobile agents are randomly selected in each sub-network at each time slot while in the "Brute Force" baseline, the optimal mobile agent selection strategies are obtained through brute force search. In particular, we first run the "Random" baseline algorithm and obtain the corresponding average payment. Taking the average payment of the "Random" baseline as the budget, the performances of Algorithm 7 and the "Brute Force" baseline are obtained. Fig. 4.6 shows the AoI of Algorithm 7 and the baselines when $B = 1$. It can be observed that Algorithm 7 provides a decrease of around 30% in AoI compared to the "Random" method and achieves almost the same performance as that of the "Brute Force" method, which demonstrates its effectiveness.

Fig. 4.7 shows the average payments of Algorithm 7 and the baselines. In particular, we note that in the proposed payment mechanism, the probability of a mobile agent being selected (i.e., $p_i^s(\cdot)$ in (4.11)) is estimated based on the BS's knowledge about the true locations of the mobile agents. Such an estimate may not be accurate since the true locations are unknown. Therefore, in the "Lower Bound" counterpart, we compute the accurate probabilities of the mobile agents being selected based on their true locations. It can be observed that the payment for the proposed algorithm is almost the same as that of the lower bound.

Fig. 4.8 and Fig. 4.9 show the AoI performance and the average payment (per sub-network) of Algorithm 8 for different number of sub-networks, each with the same setting as the one

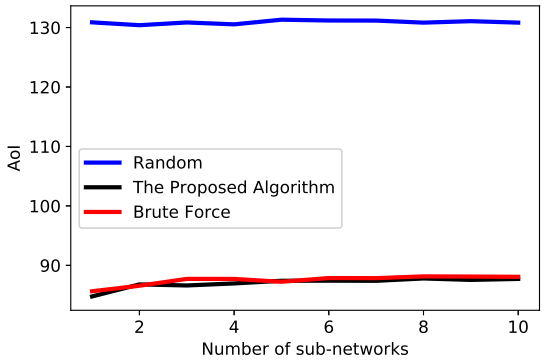

Figure 4.8: AoI Performance of Algorithm 8

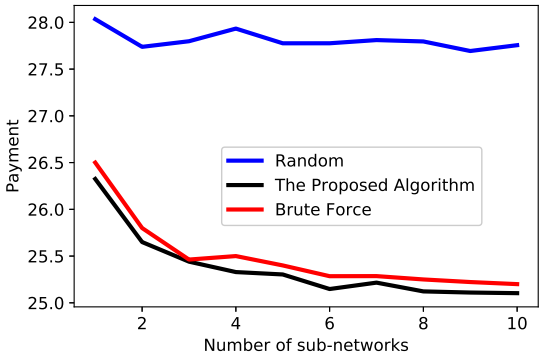

Figure 4.9: Average Payment of Algorithm 8

examined in Fig. 4.6. In particular, when $B > 1$, the budget of the BS is selected such that the AoI is the same as that in the single sub-network scenario. It can be observed that for the same AoI performance, the required payment decreases as the number of sub-networks increases. This is because, in Algorithm 8, the budget allocated to each sub-network at each time slot depends on its AoI and the sub-networks with higher AoI are allocated more budget. With the same payment, the expected AoI reduction in the sub-networks with higher AoI is larger than that in the sub-networks with smaller AoI. As a result, the budget is more effectively used.

4.6 Related Works

Many emerging applications require persistent monitoring over a set of locations, in which mobile agents are employed to help collect information. There are several works on the design of trajectory for AoI minimization in the single agent scenario. In particular, [25] obtains the optimal permutation of nodes for the mobile agent to visit in sequence which minimizes the AoI; [23] further considers the scenarios in which the mobility of the mobile agent is constrained by a general graph and studies the optimal trajectory over the space of all trajectories allowed on the graph. [110] designs the mobile agent's trajectory as well as energy and the service time allocation to minimize the peak AoI. However, these works consider only one mobile agent, which may not be enough in a large network. In this chapter, the AoI minimization problem in a more general setting, in which multiple location privacy-aware mobile agents are employed by the BS, is considered.

Since the introduction of Geo-indistinguishability [33], there have been many existing works (see e.g., [31, 34–38] and the references therein) which use differential privacy approaches for location privacy preservation. However, most of the existing approaches only use differential privacy as a tool for location privacy protection and assume a fixed privacy level, either determined by the users or the BS. [39] considers differential location privacy in the design of the

incentive mechanism. However, it assumes that the workers share their true location information with a trustworthy BS which releases the location information in a differentially private manner. In this sense, the workers do not have control over their own location information. With such consideration, [40] develops a trading market which allows the workers to determine their own privacy levels for location privacy. In particular, the BS first announces a set of sensing tasks to the workers, and the workers determine their privacy levels as well as sensing costs and submit bids to the BS. After the BS determines the winners, they embed the obfuscated locations in their sensing reports such that a certain level of location differential privacy is preserved. However, since auction based method is considered, the workers need to determine their privacy levels first before submitting their bids, which is not desired in the considered scenarios. In addition, the optimality of the differentially private obfuscation strategy is not established. In this chapter, a payment mechanism which can stimulate the mobile agents to select the ideal privacy levels for the BS is proposed. In addition, with the given payment mechanism, the optimality of the differentially private obfuscation strategy is proved under certain conditions.

Besides, some theoretical research efforts have been devoted to the study of private data (other than location privacy) trading in terms of differential privacy, in which the users can determine their own privacy levels given the payment. [111] designs a coupon game in which the agent is offered a coupon for the signal it sends. The value of the coupon determines the level of privacy that the agent will select. [106, 107] propose a game among the data sellers and the optimal payment to incentivize the workers to report with certain privacy level is established by solving the Nash equilibrium of the game. However, on the one hand, these works only consider binary data, which is not the case in the considered location privacy scenario. On the other hand, their approaches cannot be directly used in the considered AoI minimization problem in which the moving trajectories of the mobile agents are also taken into account.

There are also some differential privacy-preserving incentive mechanisms concerning the privacy of the workers' bids or sensing data. [112–114] allow the workers to obfuscate their bids or sensing data before sending them to the BS. However, in these works, the privacy levels are determined by the BS rather than the workers (i.e., the workers cannot choose their own privacy levels). [115, 116] propose privacy-preserving mechanisms to protect the workers' privacy against the outsiders. Similar to [39], it is assumed that the workers share their raw data with a trustworthy BS. In this chapter, the mobile agents determine their own privacy levels, and the location privacy is protected from the BS as well.

4.7 Conclusions

In this chapter, the AoI minimization problem in the presence of multiple location privacy-aware mobile agents is considered. In particular, a payment mechanism is proposed for the

BS to motivate the mobile agents to report their locations. The optimality of the obfuscation strategy given the payment mechanism is established, which allows the BS to adjust its payment mechanism such that the mobile agents will select its desired privacy levels. Then, given the payment mechanism, a cost-efficient mobile agent selection algorithm under budget constraint is proposed. Simulations are performed to demonstrate the effectiveness of the proposed method.

Chapter 5

Stochastic-Sign SGD for Privacy-Preserving Federated Learning with Theoretical Guarantees

In this chapter, we present Stochastic-Sign SGD, which is a class of stochastic-sign based SGD algorithms. In particular, we first propose a stochastic compressor *sto-sign*, which extends SIGNSGD to its stochastic version STO-SIGNSGD. In this scheme, instead of directly transmitting the signs of gradients, the workers adopt a two-level stochastic quantization and transmit the signs of the quantized results. We note that different from the existing 1-bit stochastic quantization schemes (e.g., QSGD [43], cpSGD [47]), the proposed algorithm also uses the majority vote rule in gradient aggregation, which allows the server-to-worker communication to be 1-bit compressed and ensures robustness as well. Then, to further resolve the privacy concerns, a differentially private stochastic compressor *dp-sign* is proposed, which can accommodate the requirement of (ϵ, δ)-local differential privacy [117]. The corresponding algorithm is termed as DP-SIGNSGD. We then prove that when the number of workers is large enough, both of the proposed algorithms converge with a rate of $O(\frac{\sqrt{d}}{\sqrt{T}})$ under heterogeneous data distribution, where d is the dimension of the hypothesis vector and T is the total number of training iterations. We also extend the proposed algorithm to its error-feedback variant, termed as Error-Feedback Stochastic-Sign SGD. In this scheme, the server keeps track of the error induced by the majority vote operation and compensates for the error in the next communication round. Both the convergence and the Byzantine resilience are established. Assuming that there are M normal (benign) workers, it is shown that the Byzantine resilience of the proposed algorithms is upper bounded by

$|\sum_{m=1}^{M}(\boldsymbol{g}_m^{(t)})_i|/b_i, \forall i$, where $(\boldsymbol{g}_m^{(t)})_i$ is the i-th entry of worker m's gradient at iteration t and $b_i \geq \max_m(\boldsymbol{g}_m^{(t)})_i$ is some design parameter. Particularly, b_i depends on the data heterogeneity (through $\max_m(\boldsymbol{g}_m^{(t)})_i$). Extensive simulations are performed to demonstrate the effectiveness of all the proposed algorithms.

The remainder of this chapter is organized as follows. Section 5.1 formulates the federated learning problem. The Stochastic-Sign SGD algorithms and the convergence analysis are presented in Section 5.2. Section 5.3 shows the Byzantine resilience of the Stochastic-Sign SGD algorithms. The discussion about the extension to SGD is presented in Section 5.4. The error-feedback variants are discussed in Section 5.5. Related works are discussed in Section 5.6, and Section 5.7 concludes this chapter.

5.1 Problem Formulation

In this chapter, we consider a typical federated optimization problem with M normal workers as in [1]. Formally, the goal is to minimize the finite-sum objective of the form

$$\min_{w\in\mathbb{R}^d} F(w) \quad \text{where} \quad F(w) \stackrel{\text{def}}{=} \frac{1}{M}\sum_{m=1}^{M} f_m(w). \tag{5.1}$$

For a machine learning problem, we have a sample space $I = X \times Y$, where X is a space of feature vectors and Y is a label space. Given the hypothesis space $\mathcal{W} \subseteq \mathbb{R}^d$, we define a loss function $l : \mathcal{W} \times I \to \mathbb{R}$ which measures the loss of prediction on the data point $(x, y) \in I$ made with the hypothesis vector $w \in \mathcal{W}$. In such a case, $f_m(w)$ is a local function defined by the local dataset of worker m and the hypothesis w. More specifically,

$$f_m(w) = \frac{1}{|D_m|}\sum_{(x_n,y_n)\in D_m} l(w;(x_n,y_n)), \tag{5.2}$$

where $|D_m|$ is the size of worker m's local dataset D_m. In many FL applications, the local datasets of the workers are heterogeneously distributed. In this case, we have $D_m \neq D_j$ and therefore $\nabla f_m(w) \neq \nabla f_j(w), \forall m \neq j$.

We consider a parameter server paradigm. At each communication round t, each worker m forms a batch of training samples, based on which it computes and transmits the stochastic gradient $\boldsymbol{g}_m^{(t)}$ as an estimate to the true gradient $\nabla f_m(w_m^{(t)})$. When the worker m evaluates the gradient over its whole local dataset, we have $\boldsymbol{g}_m^{(t)} = \nabla f_m(w_m^{(t)})$. After receiving the gradients from the workers, the server performs aggregation and sends the aggregated gradient back to the workers. Finally, the workers update their local model weights using the aggregated gradient. In this sense, the classic stochastic gradient descent (SGD) algorithm [118] performs iterations

Algorithm 9 Stochastic-Sign SGD with majority vote

1: **Input**: learning rate η, current hypothesis vector $w^{(t)}$, M workers each with an independent gradient $\boldsymbol{g}_m^{(t)}$, the 1-bit compressor $q(\cdot)$.
2: **on server:**
3: **pull** $q(\boldsymbol{g}_m^{(t)})$ **from** worker m.
4: **push** $\tilde{\boldsymbol{g}}^{(t)} = sign\big(\frac{1}{M}\sum_{m=1}^{M} q(\boldsymbol{g}_m^{(t)})\big)$ **to** all the workers.
5: **on each worker:**
6: **update** $w^{(t+1)} = w^{(t)} - \eta\tilde{\boldsymbol{g}}^{(t)}$.

of the form

$$w_m^{(t+1)} = w_m^{(t)} - \frac{\eta}{M}\sum_{m=1}^{M} \boldsymbol{g}_m^{(t)}. \tag{5.3}$$

In this case, since all the workers adopt the same update rule using the aggregated gradient, $w_m^{(t)}$'s are the same for all the workers. Therefore, in the following discussions, we omit the worker index m for the ease of presentation. To accommodate the requirement of communication efficiency in FL, we adopt the popular idea of gradient quantization and assume that each worker m quantizes the gradient with a stochastic 1-bit compressor $q(\cdot)$ and sends $q(\boldsymbol{g}_m^{(t)})$ instead of its actual local gradient $\boldsymbol{g}_m^{(t)}$. Combined with the idea of majority vote in [45], the corresponding algorithm is presented in Algorithm 9.

Intuitively, the performance of Algorithm 9 is limited by the probability of wrong aggregation, which is given by

$$sign\left(\frac{1}{M}\sum_{m=1}^{M} q(\boldsymbol{g}_m^{(t)})\right) \neq sign\left(\frac{1}{M}\sum_{m=1}^{M} \nabla f_m(w^{(t)})\right). \tag{5.4}$$

In SIGNSGD, $q(\boldsymbol{g}_m^{(t)}) = sign(\boldsymbol{g}_m^{(t)})$ and (5.4) holds when $\nabla f_m(w^{(t)}) \neq \nabla f_j(w^{(t)}), \forall m \neq j$ with a high probability, which prevents its convergence. In this chapter, we propose two compressors *sto-sign* and *dp-sign*, which guarantee that (5.4) holds with a probability that is strictly smaller than 0.5 and therefore the convergence of Algorithm 9 follows. Moreover, *dp-sign* is differentially private, i.e., given the quantized gradient $q(\boldsymbol{g}_m^{(t)})$, the adversary cannot distinguish the local dataset of worker m from its neighboring datasets that differ in only one data point with a high probability.

In addition to the M normal workers, it is assumed that there exist B Byzantine attackers, and its set is denoted as $\mathcal{B}$. Instead of using *sto-sign* and *dp-sign*, the Byzantine attackers can use an arbitrary compressor denoted by *byzantine-sign*. In this chapter, we consider the scenario that the Byzantine attackers have access to the average gradients of all the M normal workers (i.e., $\boldsymbol{g}_j^{(t)} = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{g}_m^{(t)}, \forall j \in \mathcal{B}$) and follow the same procedure as the normal workers.

Therefore, we assume that the Byzantine attacker j shares the opposite signs of the true gradients, i.e., $byzantine\text{-}sign(\boldsymbol{g}_j^{(t)}) = -sign(\boldsymbol{g}_j^{(t)})$.

In order to facilitate the convergence analysis, the following commonly adopted assumptions are made.

Assumption 4. *(Lower bound). For all w and some constant F^*, we have objective value $F(w) \geq F^*$.*

Assumption 5. *(Smoothness). $\forall w_1, w_2$, we require for some non-negative constant L*

$$F(w_1) \leq F(w_2) + < \nabla F(w_2), w_1 - w_2 > + \frac{L}{2}||w_1 - w_2||_2^2, \tag{5.5}$$

where $< \cdot, \cdot >$ is the standard inner product.

Assumption 6. *(Variance bound). For any worker m, the stochastic gradient oracle gives an independent unbiased estimate g_m that has coordinate bounded variance:*

$$\mathbb{E}[g_m] = \nabla f_m(w), \mathbb{E}[((g_m)_i - \nabla f_m(w)_i)^2] \leq \sigma_i^2, \tag{5.6}$$

for a vector of non-negative constants $\bar{\sigma} = [\sigma_1, \cdots, \sigma_d]$; $(g_m)_i$ and $\nabla f_m(w)_i$ are the i-th coordinate of the stochastic and the true gradient, respectively.

Assumption 7. *The total number of workers is odd.*

We note that Assumptions 4, 5 and 6 are standard for non-convex optimization and Assumption 7 is just to ensure that there is always a winner in the majority vote [49], which can be easily relaxed.

Experimental Settings. To facilitate empirical discussions on our proposed algorithms in the remaining sections, we first introduce our experimental settings here. We implement our proposed method with a two-layer fully connected neural network on the standard MNIST dataset [119]. We consider a scenario of $M = 31$ normal workers. To simulate the heterogeneous data distribution scenario, each worker only stores exclusive data for one out of the ten categories, unless otherwise noted. We use a constant learning rate and tune the parameter from the set $\{1, 0.1, 0.01, 0.005, 0.001, 0.0001\}$. We compare the proposed algorithms with two baselines: SIGNSGD [45] and FedAvg [1]. More details about the implementation can be found in Appendix D.4.

5.2 Algorithms and Convergence Analysis

In this section, we propose two compressors *sto-sign* and *dp-sign* for the Stochastic-Sign SGD framework, which can deal with the heterogeneous data distribution scenario. The basic ideas of

the two compressors are given as follows.

- *sto-sign*: instead of directly sharing the signs of the gradients, *sto-sign* first performs a two-level stochastic quantization and then transmits the signs of the quantized results.
- *dp-sign*: it is a differentially private version of *sto-sign*. The probability of each coordinate of the gradients mapping to $\{-1, 1\}$ is designed to accommodate the local differential privacy requirements.

In this section, we first consider the scenario in which all the workers are benign. The Byzantine resilience of *sto-sign* and *dp-sign* will be discussed in Section 5.3. In addition, we assume that each worker evaluates the gradients over its whole local dataset for simplicity (i.e., $\boldsymbol{g}_m^{(t)} = \nabla f_m(w^{(t)}), \forall 1 \leq m \leq M$). Particularly, in federated learning, the workers usually compute $\nabla f_m(w^{(t)})$ due to the small size of the local dataset. The discussion about stochastic gradients is presented in Section 5.4.

5.2.1 The Stochastic Compressor *sto-sign*

Formally, the compressor *sto-sign* is defined as follows.

Definition 9. *For any given gradient $\boldsymbol{g}_m^{(t)}$, the compressor sto-sign outputs sto-sign$(\boldsymbol{g}_m^{(t)}, \boldsymbol{b})$, where $\boldsymbol{b}$ is a vector of design parameters. The i-th entry of sto-sign$(\boldsymbol{g}_m^{(t)}, \boldsymbol{b})$ is given by*

$$\textit{sto-sign}(\boldsymbol{g}_m^{(t)}, \boldsymbol{b})_i = \begin{cases} 1, \ \textit{with probability } \frac{b_i + (\boldsymbol{g}_m^{(t)})_i}{2b_i}, \\ -1, \textit{with probability } \frac{b_i - (\boldsymbol{g}_m^{(t)})_i}{2b_i}, \end{cases} \tag{5.7}$$

where $(\boldsymbol{g}_m^{(t)})_i$ and $b_i \geq max_m |(\boldsymbol{g}_m^{(t)})_i|$ are the i-th entry of $\boldsymbol{g}_m^{(t)}$ and $\boldsymbol{b}$, respectively.

Recall that the performance of Algorithm 9 largely depends on the probability of wrong aggregation (c.f. (5.4)). When $q(\boldsymbol{g}_m^{(t)}) = sign(\boldsymbol{g}_m^{(t)})$, whether (5.4) holds or not is determined by the gradients $\boldsymbol{g}_m^{(t)}$'s, which are unknown. As a result, the convergence of SIGNSGD is not guaranteed. The key idea of *sto-sign* is to introduce the stochasticity such that the probability of wrong aggregation can be theoretically bounded for an arbitrary realization of $\boldsymbol{g}_m^{(t)}$'s.

In the following discussion, we term Algorithm 9 with $q(\boldsymbol{g}_m^{(t)}) = \textit{sto-sign}(\boldsymbol{g}_m^{(t)}, \boldsymbol{b})$ as STO-SIGNSGD. For the ease of presentation, we consider the scalar case and obtain the following results for the compressor *sto-sign*. They can be readily generalized to the vector case by applying the results independently on each coordinate.

Theorem 13. *Let $u_1, u_2, \cdots, u_M$ be M known and fixed real numbers and consider random variables $\hat{u}_m = sto\text{-}sign(u_m, b)$, $1 \leq m \leq M$. Then we have*

$$P\left(sign\left(\frac{1}{M}\sum_{m=1}^{M}\hat{u}_m\right) \neq sign\left(\frac{1}{M}\sum_{m=1}^{M}u_m\right)\right) < \left[\Big(1-x\Big)e^x\right]^{\frac{M}{2}}, \tag{5.8}$$

where $x = \frac{|\sum_{m=1}^{M}u_m|}{bM}$.

Proof. Please see Appendix D.1.1. Here we provide some intuition about the proof. Given the majority vote rule, the aggregation result is wrong if more than half of the workers share the wrong signs. In addition, based on (5.7), we can obtain the probability of each worker sharing 1 or -1. Therefore, the number of workers that share the wrong signs can be modeled as a Poisson binomial variable, denoted as Z. The key difficulty is that the correct sign $sign(\frac{1}{M}\sum_{m=1}^{M}u_m)$ is unknown. However, thanks to the special structure of (5.7), the mean of the number of workers sharing either -1 or 1 depends on $\frac{1}{M}\sum_{m=1}^{M}u_m$ rather than on each individual u_m. That being said, we can always obtain the expectation of Z as a function of $\frac{1}{M}\sum_{m=1}^{M}u_m$. As a result, we can invoke the Markov inequality and obtain (5.8) after some algebra. □

Remark 16. ***(selection of $\boldsymbol{b}$)*** *Some discussions on the choice of the vector $\boldsymbol{b}$ in (5.7) are in order. We take the i-th entry of $\boldsymbol{b}$ as an example. In the FL application, the i-th entry of the gradient $\boldsymbol{g}_m^{(t)}$ corresponds to u_m in Theorem 13. On the one hand, according to the definition of sto-sign, $b_i \geq max_m|(\boldsymbol{g}_m^{(t)})_i|$. On the other hand, it can be shown that $\Big(1-x\Big)e^x$ in (5.8) is a decreasing function of x (and therefore an increasing function of b_i) when $x < 1$. Therefore, to minimize the probability of wrong aggregation, it is optimal to select $b_i = max_m|(\boldsymbol{g}_m^{(t)})_i|$. In addition, since the true gradients change during the training process, the optimal b_i varies across the iterations too. In the implementation of sto-sign, for a fixed vector $\boldsymbol{b}$, it is possible that $b_i < max_m|(\boldsymbol{g}_m^{(t)})_i|$ for some coordinates. In such cases, the probabilities defined in (5.7) may fall out of the range $[0, 1]$. We round them to 1 if they are positive and 0 otherwise. However, in practice, since $max_m|(\boldsymbol{g}_m^{(t)})_i|$ is unknown, the selection of an appropriate $\boldsymbol{b}$ is an interesting problem deserving further investigation.*

Theorem 14. *Given the same $\{u_m\}_{m=1}^M$ and $\{\hat{u}_m\}_{m=1}^M$ as those in Theorem 13, for a sufficiently large b, we have $P\Big(sign\Big(\frac{1}{M}\sum_{m=1}^{M}\hat{u}_m\Big) \neq sign\Big(\frac{1}{M}\sum_{m=1}^{M}u_m\Big)\Big) < \frac{1}{2}$.*

Proof. Please see Appendix D.1.2. □

Theorem 15. *Suppose Assumptions 4, 5 and 7 are satisfied, and set the learning rate $\eta = \frac{1}{\sqrt{Td}}$.*

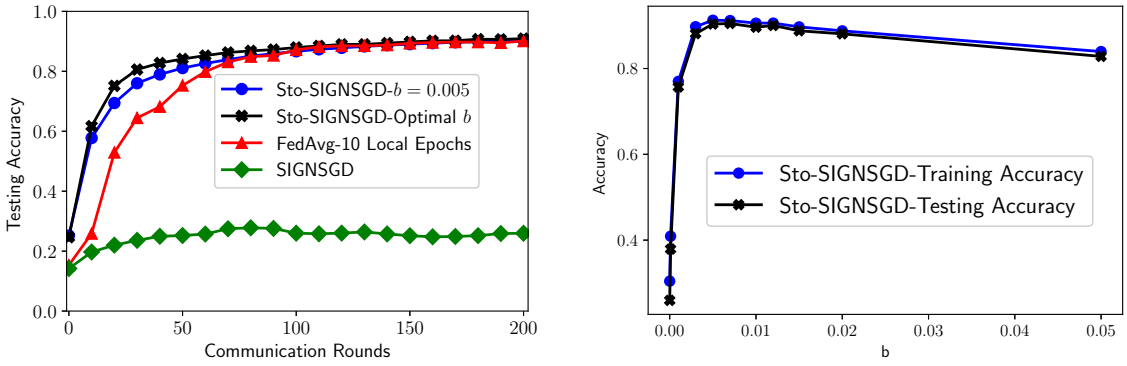

Figure 5.1: The left figure compares the training accuracy of Sto-SIGNSGD with SIGNSGD and FedAvg [1]. The right figure shows the training and the testing accuracy of Sto-SIGNSGD for different $\boldsymbol{b} = b \cdot \mathbf{1}$. The results are averaged over 5 repeats. For FedAvg, we tune the number of local epochs from the set {1, 10, 20, 30} and present the best results.

Then for any M, by running Sto-SIGNSGD for T iterations, we have

$$\frac{1}{T}\sum_{t=1}^{T} c||\nabla F(w^{(t)})||_1 \leq \frac{(F(w^{(0)}) - F^*)\sqrt{d}}{\sqrt{T}} + \frac{L\sqrt{d}}{2\sqrt{T}} + 2bd\Delta(M), \tag{5.9}$$

where $0 < c < 1$ is some positive constant, and $\Delta(M)$ is the solution to $\left[\left(1-x\right)e^{x}\right]^{\frac{M}{2}} = \frac{1-c}{2}$.

Proof. Please see Appendix D.1.3. □

Remark 17. *Similar to SIGNSGD, the convergence rate of Sto-SIGNSGD depends on the L_1-norm of the gradient. A detailed discussion on this feature can be found in [45]. Note that compared to the convergence rate of SIGNSGD, there is a positive coefficient $c < 1$. This can be understood as the cost of dealing with the heterogeneous data distribution.*

Remark 18. *It can be verified that $\Delta(M)$ is a decreasing function of M and $\lim_{M\to\infty}\Delta(M) = 0$. The last term in (5.9) captures the gap induced by the scenarios where the probability of wrong aggregation is larger than $\frac{1-c}{2}$. If we select $b \propto \frac{1}{\sqrt{dT}}$, the right hand side of (5.9) is upper bounded by $O(\frac{\sqrt{d}}{\sqrt{T}})$.*

Moreover, note that Theorem 15 holds for any b. According to Theorem 14, the probability of wrong aggregation is strictly smaller than $\frac{1}{2}$ when b is sufficiently large. That being said, there exists a positive constant c such that the probability of wrong aggregation is no larger than $\frac{1-c}{2}$. As a result, the last term in (5.9) can be eliminated in such a case. However, a larger b corresponds to a smaller positive constant c, which negatively impact the convergence.

Experimental results. We perform experiments to examine the learning performance of Sto-SIGNSGD for different selection of $\boldsymbol{b}$. Throughout our experiments, in the fixed $\boldsymbol{b}$ scenarios, we set $\boldsymbol{b} = b \cdot \mathbf{1}$ for some positive constant b. For "Optimal $\boldsymbol{b}$", we set $b_i = max_m|(\boldsymbol{g}_m^{(t)})_i|, \forall i$. The results are shown in Figure 5.1. It can be observed that Sto-SIGNSGD outperforms SIGNSGD and FedAvg, and the performance of "$b = 0.005$" is almost the same as "Optimal $\boldsymbol{b}$". That being said, compared to FedAvg, Sto-SIGNSGD achieves better performance while requires less communication overhead per communication round. In addition, it can be observed that for fixed $\boldsymbol{b}$, b should be large enough to optimize the performance. Then when b keeps increasing, both the training accuracy and the testing accuracy decrease, which corroborates our analysis above.

5.2.2 The Differentially Private Compressor *dp-sign*

In this subsection, we present the differentially private version of *sto-sign*. In this chapter, we study the privacy guarantee of the proposed algorithms from the lens of local differential privacy [117], which provides a strong notion of individual privacy in data analysis. The definition of local differential privacy is formally given as follows.

Definition 10. *Given a set of local datasets $\mathcal{D}$ provided with a notion of neighboring local datasets $\mathcal{N}_{\mathcal{D}} \subset \mathcal{D} \times \mathcal{D}$ that differ in only one data point. For a query function $f : \mathcal{D} \to \mathcal{X}$, a mechanism $\mathcal{M} : \mathcal{X} \to \mathcal{O}$ to release the answer of the query is defined to be (ϵ, δ)-locally differentially private if for any measurable subset $\mathcal{S} \subseteq \mathcal{O}$ and two neighboring local datasets $(D_1, D_2) \in \mathcal{N}_{\mathcal{D}}$,*

$$P(\mathcal{M}(f(D_1)) \in \mathcal{S}) \leq e^{\epsilon} P(\mathcal{M}(f(D_2)) \in \mathcal{S}) + \delta. \tag{5.10}$$

A key quantity in characterizing local differential privacy for many mechanisms is the sensitivity of the query f in a given norm l_r, which is defined as

$$\Delta_r = \max_{(D_1, D_2) \in \mathcal{N}_{\mathcal{D}}} ||f(D_1) - f(D_2)||_r. \tag{5.11}$$

For more details about the concept of differential privacy, the reader is referred to [117] for a survey.

Formally, the compressor *dp-sign* is defined as follows.

Definition 11. *For any given gradient $\boldsymbol{g}_m^{(t)}$, the compressor dp-sign outputs dp-sign$(\boldsymbol{g}_m^{(t)}, \epsilon, \delta)$. The i-th entry of dp-sign$(\boldsymbol{g}_m^{(t)}, \epsilon, \delta)$ is given by*

$$dp\text{-}sign(\boldsymbol{g}_m^{(t)}, \epsilon, \delta)_i = \begin{cases} 1, & \text{with probability } \Phi\Big(\frac{(\boldsymbol{g}_m^{(t)})_i}{\sigma}\Big) \\ -1, & \text{with probability } 1 - \Phi\Big(\frac{(\boldsymbol{g}_m^{(t)})_i}{\sigma}\Big) \end{cases} \tag{5.12}$$

where $\Phi(\cdot)$ is the cumulative distribution function of the normalized Gaussian distribution; $\sigma = \frac{\Delta_2}{\epsilon}\sqrt{2\ln(\frac{1.25}{\delta})}$, where ϵ and δ are the differential privacy parameters and Δ_2 is the sensitivity measure.

Theorem 16. *The proposed compressor dp-sign$(\cdot, \epsilon, \delta)$ is (ϵ, δ)-differentially private for any $\epsilon, \delta \in (0, 1)$.*

Proof. Please see Appendix D.1.4. □

Remark 19. *Note that throughout this chapter, we assume $\delta > 0$. For the $\delta = 0$ scenario, the Laplace mechanism [117] can be used by replacing the cumulative distribution function of the normalized Gaussian distribution in (5.12) with that of the Laplace distribution. The corresponding discussion can be found in Appendix D.2.*

We term Algorithm 9 with $q(\boldsymbol{g}_m^{(t)}) = dp\text{-}sign(\boldsymbol{g}_m^{(t)}, \epsilon, \delta)$ as DP-SIGNSGD. Similar to *sto-sign*, we consider the scalar case and obtain the following result for $dp\text{-}sign(\cdot, \epsilon, \delta)$.

Theorem 17. *Let $u_1, u_2, \cdots, u_M$ be M known and fixed real numbers. Further define random variables $\hat{u}_i = dp\text{-}sign(u_i, \epsilon, \delta), \forall 1 \leq i \leq M$. Then there always exist a constant σ_0 such that when $\sigma \geq \sigma_0$, $P(sign(\frac{1}{M}\sum_{m=1}^{M}\hat{u}_i) \neq sign(\frac{1}{M}\sum_{m=1}^{M}u_i)) < \left[\left(1-x\right)e^x\right]^{\frac{M}{2}}$, where $x = \frac{|\sum_{m=1}^{M}u_m|}{\sigma M}$.*

Proof. Please see Appendix D.1.5. □

Given Theorem 17, the convergence of DP-SIGNSGD can be obtained by following a similar analysis to that of Theorem 15.

5.3 Byzantine Resilience

In this section, the Byzantine resilience of the proposed algorithms is investigated. We note that the convergence of Sto-SIGNSGD and DP-SIGNSGD is limited by the probability of wrong aggregation (i.e., more than half of the workers share the wrong signs). Let Z_i denote the number of normal workers that share (quantized) gradients with different signs from the true gradient $\nabla F(w^{(t)})$ on the i-th coordinate (i.e., $q(\boldsymbol{g}_m^{(t)})_i \neq sign(\nabla F(w^{(t)})_i)$). Then, Z_i is a Poisson binomial variable. In order to tolerate k_i Byzantine workers on the i-th coordinate of the gradient, we need to have $P(Z_i \geq \frac{M-k_i}{2}) < \frac{1}{2}$, where M is the number of benign workers. Therefore, we can prove the following theorem.

Theorem 18. *There exists a positive constant s_0 such that when $s > s_0$, Sto-SIGNSGD and DP-SIGNSGD can at least tolerate k_i Byzantine attackers on the i-th coordinate of the gradient at t-th iteration and k_i satisfies*

$$k_i < \frac{|\sum_{m=1}^{M}\nabla f_m(w^{(t)})_i|}{s}, \qquad \left[\left(1-x\right)e^x\right]^{\frac{M-k_i}{2}} < \frac{1}{2}, \tag{5.13}$$

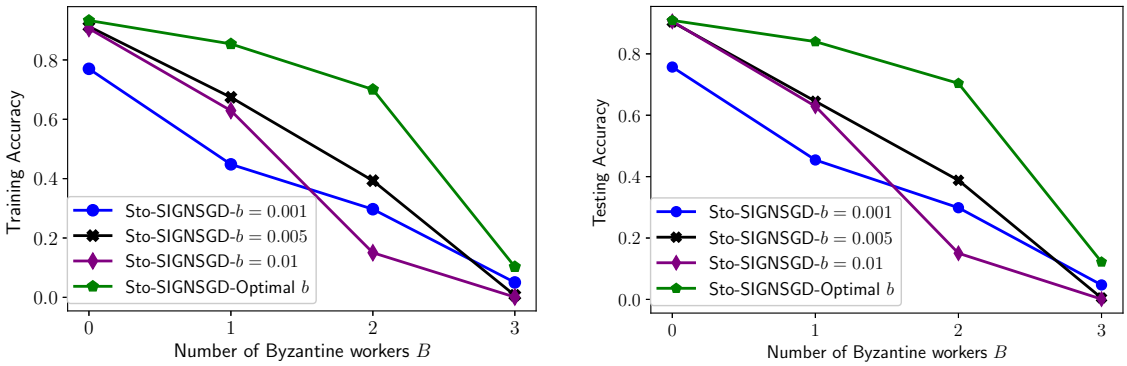

Figure 5.2: The training and the testing accuracy of Sto-SIGNSGD for different number of Byzantine workers and different $\boldsymbol{b}$.

where $x = \frac{|\sum_{m=1}^{M} \nabla f_m(w^{(t)})_i| - sk_i}{(M-k_i)s}$, $s = \sigma$ *for* DP-SIGNSGD *and* $s = b_i \geq max_m|(\boldsymbol{g}_m^{(t)})_i|$ *for* Sto-SIGNSGD.

Overall, the number of Byzantine workers that the algorithms can tolerate is given by $min_{1\leq i\leq d} k_i$.

Proof. Please see Appendix D.1.6. □

In this case, $\left[\Big(1-x\Big)e^x\right]^{\frac{M-k_i}{2}}$ measures the probability of wrong aggregation after taking the Byzantine workers into consideration. As we know, $\Big(1-x\Big)e^x$ is decreasing function of x (and therefore an increasing function of s). So the second equation of (5.13) indicates that the Byzantine tolerance decreases as s increases, which conforms to the observation from the first equation of (5.13).

According to (5.13), when *sto-sign* is used, we can set $b_i = max_m|\nabla f_m(w^{(t)})_i|$. In this case, $k_i < \frac{|\sum_{m=1}^{M} \nabla f_m(w^{(t)})_i|}{max_m|\nabla f_m(w^{(t)})_i|}$, which means that the Byzantine resilience depends on the heterogeneity of the local datasets. When the workers can access the same dataset, i.e., $\nabla f_m(w^{(t)})_i = \nabla f_n(w^{(t)})_i, \forall m, n$, Theorem 18 gives $x = 1$ and $k_i < M$. Therefore, it can tolerate $M-1$ Byzantine workers.

Remark 20. *Our analysis of the convergence and the Byzantine resilience is based on each individual coordinate of the gradients, which corresponds to the generalized Byzantine attacks and the dimensional Byzantine resilience [120]. Furthermore, it also indicates that the parameter* σ *in dp-sign can be different across coordinates and iterations, which allows one to select suitable parameters for different coordinates and iterations to improve the privacy performance of the algorithm. A similar idea has been explored in [121] without considering quantization.*

Table 5.1: Testing Accuracy of Sto-signSGD

B	1 Label	2 Labels	4 Labels
0	91.90%	94.57%	91.79%
1	85.39%	94.21%	92.42%
2	70.04%	83.95%	93.76%
3	10.31%	80.61%	92.35%
4	0.25%	57.52%	85.92%

Experimental results. Fig. 5.2 shows the performance of Sto-signSGD for different selection of $\boldsymbol{b} = b \cdot \mathbf{1}$ and different number of Byzantine workers B. It can be seen that when $b = 0.001$, it is not large enough to optimize the performance according to our results in Section 5.2. Setting $b = 0.005$ and $b = 0.01$ achieves almost the same performance as "Optimal $\boldsymbol{b}$" when there is no Byzantine worker ($B = 0$). However, as the number of Byzantine workers increases, both the training and the testing accuracy of "$b = 0.01$" drop much faster than those of "$b = 0.005$", which conforms to our analysis above that a lager b results in worse Byzantine resilience.

To examine the impact of data heterogeneity, we vary the number of labels of each worker's local training dataset. Table 5.1 shows the testing accuracy of Sto-signSGD with optimal $\boldsymbol{b}$. It can be observed that the Byzantine resilience of Sto-signSGD increases as the number of labels increases. Up to now, we examine the performance of Sto-signSGD, the results for DP-signSGD are deferred to Section 5.5.

5.4 Extending to SGD

Up until this point in the chapter, the discussions are based on the assumption that each worker can evaluate its local true gradient $\nabla f_m(w^{(t)})$ for the ease of presentation. In the SGD scenario, we have to further account for the sampling noise. Particularly, the following theorem for Sto-signSGD can be proved. The corresponding result for DP-signSGD can be obtained following a similar strategy.

Theorem 19. *Suppose Assumptions 4-7 are satisfied, and set the learning rate $\eta = \frac{1}{\sqrt{Td}}$. Then, when $\boldsymbol{b} = b \cdot \mathbf{1}$ and b is sufficiently large, Sto-signSGD converges to the (local) optimum with a rate of $O(\frac{1}{\sqrt{T}})$ if either of the following two conditions is satisfied.*

- $P\Big(sign(\frac{1}{M}\sum_{m=1}^{M}(\boldsymbol{g}_m^t)_i) \neq sign(\nabla F(w^t)_i\Big) < 0.5, \forall 1 \leq i \leq d.$
- *The mini-batch size of stochastic gradient at each iteration is at least T.*

Proof. Please see Appendix D.1.7. □

Remark 21. *Note that the first condition is not hard to satisfy. One sufficient condition is that the sampling noise of each worker is symmetric with zero mean. This assumption is also used in [48], which shows that the sampling noise is approximately not only symmetric, but also unimodal.*

Remark 22. *We note that by replacing the compressor sign in* SIGN*SGD with sto-sign or dp-sign, we can obtain the improved rate (a factor of* $\frac{1}{\sqrt{M}}$ *in the variance term) without assuming unimodal and symmetric stochastic gradient sampling noise as in [45].*

Remark 23. *We note that the above discussion assumes that* b *is sufficiently large, which guarantees that the probability of wrong aggregation is less than 0.5. For an arbitrary* b *that satisfies the condition in the definition of sto-sign, we believe that it is possible to prove that the algorithm converges to the neighborhood of the (local) optimum. In particular, similar to the proof of Theorem 15, there will be an additional term* $\sum_{i=1}^{d} |\nabla F(w^t)_i| \mathbb{1}_{|\frac{1}{M}\sum_{m=1}^{M}(\boldsymbol{g}_m^t)_i| \leq b\Delta(M)}$*. It is possible to upper bound this additional term given the fact that* $\mathbb{E}[\frac{1}{M}\sum_{m=1}^{M}(\boldsymbol{g}_m^t)_i] = \nabla F(w^t)_i$*, despite that more efforts are required to make the analysis rigorous.*

5.5 Extending to Error-feedback Variant

To further improved the performance of Algorithm 9, we incorporate the error-feedback technique and propose its error-feedback variant, which is presented in Algorithm 10.

Algorithm 10 Error-Feedback Stochastic-Sign SGD with majority vote

1: **Input**: learning rate η, current hypothesis vector $w^{(t)}$, current residual error vector $\tilde{\boldsymbol{e}}^{(t)}$, M workers each with an independent gradient $\boldsymbol{g}_m^{(t)} = \nabla f_m(w^{(t)})$, the 1-bit compressor $q(\cdot)$.
2: **on server:**
3: **pull** $q(\boldsymbol{g}_m^{(t)})$ **from** worker m.
4: **push** $\tilde{\boldsymbol{g}}^{(t)} = sign\big(\frac{1}{M}\sum_{m=1}^{M} q(\boldsymbol{g}_m^{(t)}) + \tilde{\boldsymbol{e}}^{(t)}\big)$ **to** all the workers,
5: **update residual error:**

$$\tilde{\boldsymbol{e}}^{(t+1)} = \frac{1}{M}\sum_{m=1}^{M} q(\boldsymbol{g}_m^{(t)}) + \tilde{\boldsymbol{e}}^{(t)} - \frac{1}{M}\tilde{\boldsymbol{g}}^{(t)}. \tag{5.14}$$

6: **on each worker:**
7: **update** $w^{(t+1)} = w^{(t)} - \eta\tilde{\boldsymbol{g}}^{(t)}$.

Remark 24. *Note that in Algorithm 10, only the server adopts the error-feedback method. When dp-sign is used, implementing error-feedback on the worker's side may increase the privacy*

leakage. Accounting for the additional privacy leakage caused by error-feedback is left as future work.

Remark 25. *In (5.14), by adding the coefficient $\frac{1}{M}$ to $\tilde{\boldsymbol{g}}^{(t)}$, the server keeps the magnitude information about the aggregation results and enables more effective error-feedback performance. More discussion about the parameter $\frac{1}{M}$ is can be found in Appendix D.3.*

Both *sto-sign* and *dp-sign* can be used in Algorithm 10 and the corresponding algorithms are termed as EF-Sto-signSGD and EF-DP-signSGD, respectively. In the following, we show the convergence and Byzantine resilience of Algorithm 10 when *dp-sign* is used. The results can be easily adapted for *sto-sign*. Particularly, the following theorems can be proved.

Theorem 20. *When Assumptions 4, 5 and 7 are satisfied, there exists a σ_0 such that when $\sigma \geq \sigma_0$, by running Algorithm 10 with $\eta = \frac{1}{M\sqrt{Td}}$, we have*

$$\frac{1}{T}\sum_{t=0}^{T-1}\frac{||\nabla F(w^{(t)})||_2^2}{\sigma} \leq \frac{(F(w_0)-F^*)\sqrt{d}}{\sqrt{T}} + \frac{(1+L+L^2\beta)\sqrt{d}}{\sqrt{T}}, \tag{5.15}$$

where β is some positive constant.

Proof. Please see Appendix D.1.8. □

Besides the fact that error-feedback is only used on the server's side, another difference between Algorithm 10 and those in [122, 123] is that it does not require the workers to share the magnitude information about the gradients. On the one hand, it saves communication overhead. On the other hand, it keeps the resilience against the re-scaling attacks. By following a similar strategy to the proofs of Theorem 20 and considering the impact of Byzantine attackers, we obtain the Byzantine resilience of Algorithm 10 as follows.

Theorem 21. *At each iteration t, there exists a constant σ_0 such that when $\sigma > \sigma_0$, Algorithm 10 can at least tolerate $k_i = |\sum_{m=1}^{M}\nabla f_m(w^{(t)})_i|/\sigma$ Byzantine attackers on the i-th coordinate of the gradient. Overall, the number of Byzantine workers that Algorithm 10 can tolerate is given by $min_{1\leq i\leq d}k_i$.*

Proof. Please see Appendix D.1.9. □

Experimental results. For DP-signSGD and EF-DP-signSGD, we follow the idea of gradient clipping in [124] to bound the sensitivity Δ_2. After computing the gradient for each individual training sample in the local dataset, each worker clips it in its L_2 norm for a clipping threshold C to ensure that $\Delta_2 \leq C$. We set $C = 4$ in the experiments and the results are shown in Fig. 5.3. It can be observed from the first two figures that when there is no Byzantine attackers, EF-DP-signSGD outperforms DP-signSGD for all the examined ϵ's, which demonstrates its effectiveness.

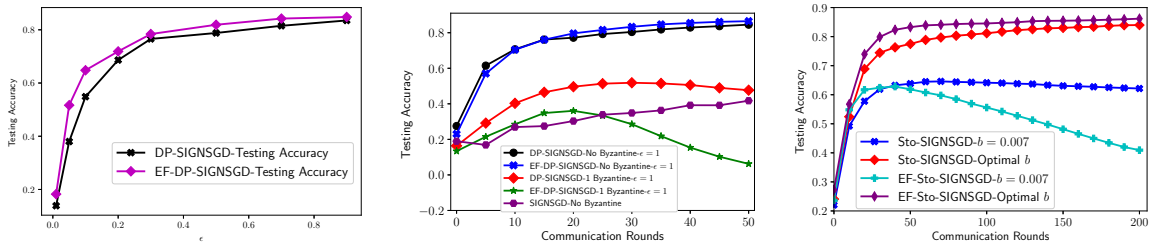

Figure 5.3: The first figure shows the performance of DP-SIGNSGD and EF-DP-SIGNSGD for different ϵ when $\delta = 10^{-5}$, without Byzantine attackers. The ϵ's measure the per epoch privacy guarantee of the algorithms. The second figure compares EF-DP-SIGNSGD with DP-SIGNSGD when $\epsilon = 1$. The last figure compares Sto-SIGNSGD with EF-Sto-SIGNSGD in the presence of 1 Byzantine attacker.

In addition, both DP-SIGNSGD and EF-DP-SIGNSGD outperform SIGNSGD, while providing privacy guarantees.

Another observation is that the error-feedback variants do not necessarily perform better. For instance, in the second figure of Fig. 5.3, when there is one Byzantine attacker and $\epsilon = 1$, the testing accuracy of EF-DP-SIGNSGD is worse than that of DP-SIGNSGD. In the beginning of the training process, k_i's in Theorem 21 are large enough such that the algorithm can tolerate the Byzantine attacker. As the gradients decrease, the probability of wrong aggregation increases. In this case, the error-feedback mechanism may carry the wrong aggregations to the future iterations and have a negative impact on the learning process. Similar results are obtained for Sto-SIGNSGD when $b = 0.007$ in the last figure of Fig. 5.3. In the meantime, for "Optimal $\boldsymbol{b}$", the error-feedback variant can tolerate the Byzantine attacker and therefore provide better performance.

5.6 Related Works

Gradient Quantization: To accommodate the need of communication efficiency in distributed learning, various gradient compression methods have been proposed. Most of the existing works focus on unbiased methods [125, 126]. QSGD [43], TernGrad [44] and ATOMO [127] propose to use stochastic quantization schemes, based on which a differentially private variant is proposed in [47]. Due to the unbiased nature of such quantization methods, the convergence of the corresponding algorithms can be established.

The idea of sharing the signs of gradients in SGD can be traced back to 1-bit SGD [63]. [128] and [45, 48] show theoretical and empirical evidence that sign based gradient schemes can converge well despite the biased approximation nature in the homogeneous data distribution scenario. In the heterogeneous data distribution case, [49] shows that the convergence of SIGNSGD

is not guaranteed and proposes to add carefully designed noise to ensure a convergence rate of $O(d^{\frac{3}{4}}/T^{\frac{1}{4}})$. However, their analysis assumes second order differentiability of the noise probability density function and cannot be applied to some commonly used noise distributions (e.g., uniform and Laplace distributions). In addition, their analysis requires that the variance of the noise goes to infinity as the number of iterations grows, which may be unrealistic in practice.

Error-Compensated SGD: Instead of directly using the biased approximation of the gradients, [63] corrects the quantization error by adding error feedback in subsequent updates and observes almost no accuracy loss empirically. [46] proposes the error-compensated quantized SGD in quadratic optimization and proves its convergence for unbiased stochastic quantization. [129] proves the convergence of the proposed error compensated algorithm for strongly-convex loss functions and [130] proves the convergence of sparsified gradient methods with error compensation for both convex and non-convex loss functions. In addition, [122] proposes EF-SIGNSGD, which combines the error compensation methods and SIGNSGD; however, only the single worker scenario is considered. [123] further extends it to the multi-worker scenario and the convergence is established. However, it is required in these two works that the compressing error cannot be larger than the magnitude of the original vector, which is not the case for some biased compressors like SIGNSGD. [131] considers more general compressors and proves the convergence under the assumption that the compressors have bounded magnitude of error. However, to the best our knowledge, none of the existing works consider the Byzantine resilience of the error-compensated methods.

Byzantine Tolerant SGD in Heterogenous Environment: There have been significant research interests in developing SGD based Byzantine tolerant algorithms, most of which consider homogeneous data distribution, e.g., Krum [132], ByzantineSGD [133], and the median based algorithms [134]. [48] shows that SIGNSGD can tolerate up to half "blind" Byzantine workers who determine how to manipulate their gradients before observing the gradients.

To accommodate the need for robust FL, some Byzantine tolerant algorithms that can deal with heterogeneous data distribution have been developed. [135] proposes to incorporate a regularized term with the objective function. However, it requires strong convexity and can only converge to the neighborhood of the optimal solution. [136] uses trimmed mean to aggregate the shared parameters. Nonetheless, it can only tolerate a small (unknown) number of Byzantine workers. In addition, both [135] and [136] assume model aggregation, i.e., both the workers and the parameter server share their models with others in full precision, which may incur significant communication cost.

5.7 Conclusions

We propose a Stochastic-Sign SGD framework that utilizes two novel gradient compressors and can deal with heterogeneous data distribution. The proposed algorithms are proved to converge in the heterogeneous data distribution scenario. In particular, the proposed differentially private compressor *dp-sign* improves the privacy and the accuracy simultaneously without sacrificing any communication efficiency. Then, we further improve the learning performance of the proposed method by incorporating the error-feedback scheme. In addition, the Byzantine resilience of the proposed algorithms is shown analytically. It is expected that the proposed algorithms can find wide applications in the design of communication efficient, differentially private, and Byzantine resilient FL algorithms.

Chapter 6

Communication Efficient Federated Learning with Energy Awareness over Wireless Networks

In Chapter 5, it is assumed that the communications between the workers and the parameter server are error-free. However, such an assumption may not hold in practice, especially over wireless links. In this chapter, the practical implementation of the sign based SGD algorithms over wireless networks is investigated. The workers are assumed to transmit their parameter updates over flat-fading channels and CSI is only available at the receiver side. Channel capacity with outage is considered and each worker is supposed to determine its transmission rate and transmission power. The impact of wireless communications on the performance of SIGNSGD over wireless networks is first analyzed. In this case, the learning performance depends on the number of communication rounds that the FL algorithm can be run and the outage probabilities of the workers for each communication round. Intuitively, increasing the transmission power and decreasing the transmission rate of a worker both decrease its outage probability. However, increasing the transmission power results in higher energy consumption for communication while decreasing the transmission rate requires faster local computation (i.e., training the local FL model) given a fixed time duration for each communication round, which leads to higher energy consumption for local computation. With such consideration, two optimization problems are formulated and solved. The first problem minimizes the energy consumption of the workers given the learning performance requirement, while the second problem optimizes the learning performance given the energy consumption requirement. Then, the scenario with heterogeneous data distribution across the workers is considered and the Stochastic-Sign SGD in Chapter 5 is adapted by considering the outage probabilities of the workers. Extensive simulations are performed to demonstrate the effectiveness of the proposed method. Particularly, compared with

SIGNSGD, the proposed stochastic sign based algorithm achieves better learning performance while reducing the energy consumption of the workers.

The remainder of this chapter is organized as follows. Section 6.1 introduces the system model. Performance analysis of SIGNSGD over wireless networks is provided in Section 6.2. The optimization problems are formulated in Section 6.3 and the corresponding solutions are presented in Section 6.4. Section 6.5 extends the proposed method to the scenario with heterogeneous data distribution across the workers. Section 6.6 presents the simulation results. Section 6.7 discusses the related works, and Section 6.8 concludes this chapter.

6.1 System Model

In this chapter, a wireless multi-user system consisting of one parameter server and a set of M workers is considered. In particular, each worker $m \in \mathcal{M}$ stores a local dataset $\mathcal{D}_m$, which will be used for local training. The local dataset can be locally generated or collected through each worker's usage of mobile devices. Considering that the training of a prediction model, especially in deep learning, usually requires a large dataset, the goal of the workers is to cooperatively learn a machine learning model while keeping the local training data on their mobile devices.

6.1.1 Machine Learning Model

A typical federated learning problem with M normal workers is considered. Formally, the goal is to minimize a finite-sum objective of the form

$$\min_{w \in \mathbb{R}^d} F(w) \quad \text{where} \quad F(w) \stackrel{\text{def}}{=} \frac{1}{M} \sum_{m=1}^{M} F_m(w). \tag{6.1}$$

For a machine learning problem, we have a sample space $I = X \times Y$, where X is a space of feature vectors and Y is a label space. Given the hypothesis space $\mathcal{W} \subseteq \mathbb{R}^d$, we define a loss function $l : \mathcal{W} \times I \to \mathbb{R}$ which measures the loss of prediction on the data point $(x, y) \in I$ made with the hypothesis vector $w \in \mathcal{W}$. In such a case, $F_m(w)$ is a local function defined by the local dataset of worker m and the hypothesis w. More specifically,

$$F_m(w) = \frac{1}{|\mathcal{D}_m|} \sum_{(x_n, y_n) \in \mathcal{D}_m} l(w; (x_n, y_n)), \tag{6.2}$$

where $|\mathcal{D}_m|$ is the size of worker m's local dataset $\mathcal{D}_m$. The loss function $l(w; (x_n, y_n))$ depends on the learning tasks and the machine learning models.

To accommodate the requirement of communication efficiency in FL, we adopt the popular idea of gradient quantization as in SIGNSGD with majority vote [45], which is presented in

Algorithm 11 SIGNSGD with majority vote [45] over wireless networks

1: Input: initial weight: $w^{(0)}$; number of workers: M; learning rate: η.
2: **for** $t = 0, 1, \cdots, T$ **do**
3: Each worker m obtains its gradient $g_m^{(t)} = \nabla F_m(w^{(t)})$ and transmits $sign(g_m^{(t)})$ to the parameter server over wireless links.
4: The parameter server obtains a noisy estimate (denoted by $\hat{g}_m^{(t)}$) of the transmitted information $sign(g_m^{(t)})$ from each worker m and sends the aggregated result $\tilde{g}^{(t)} = sign\big(\frac{1}{M}\sum_{m=1}^{M}\hat{g}_m^{(t)}\big)$ back to the workers.
5: The workers update their local models

$$w^{(t+1)} = w^{(t)} - \eta\tilde{g}^{(t)}. \tag{6.3}$$

6: **end for**

Algorithm 11. At t-th communication round, each worker m computes the gradient $g_m^{(t)}$ based on its locally stored model weights $w^{(t)}$ and the local datasets $\mathcal{D}_m$. Then, instead of transmitting the gradient $g_m^{(t)}$ directly, worker m transmits $sign(g_m^{(t)})$ to the parameter server, in which $sign(\cdot)$ is the sign function. After receiving the shared signs of the gradients from the workers (prone to channel errors), the parameter server performs aggregation using the majority vote rule and sends the aggregated result back to the workers. Finally, the workers update their local model weights using the aggregated result.

6.1.2 Local Computation Model

In this chapter, we consider a similar local computation model as those in [50] and [58]. Let c_m and f_m denote the number of CPU cycles required for worker m to process per bit data and its CPU cycle frequency, respectively, which are assumed known to the parameter server. Then, the CPU energy consumption of worker m for the local computation of one communication round is given by [137]

$$E_m^{cmp} = \frac{\alpha_m}{2} c_m D_m f_m^2, \tag{6.4}$$

in which $\frac{\alpha_m}{2}$ is the effective capacitance coefficient of worker m's computing chip, D_m is the size of worker m's training data for each communication round (in bits). In addition, the computation time for each communication round of worker m is given by

$$T_m^{cmp} = \frac{c_m D_m}{f_m}. \tag{6.5}$$

6.1.3 Transmission Model

In this chapter, it is assumed that the workers transmit their local updates (i.e., the signs of the gradients) to the parameter server via the orthogonal frequency division multiple access (OFDMA), and do not interfere with each other. Given that the parameter server has more power and bandwidth compared to the mobile devices, the downlink transmission time is ignored in this chapter.[1] Moreover, similar to most of the existing literature (e.g., [50, 58]), it is assumed that the downlink transmissions are error-free for simplicity.

For the uplink transmission, different from the existing works that consider CSI at both the transmitter and the receiver sides, we consider flat-fading channels with receiver only CSI and the capacity with outage. Capacity with outage is defined as the maximum rate that can be transmitted over a channel with a certain outage probability, which corresponds to the probability that an outage happens and the transmission cannot be decoded correctly [138]. For each worker m, we assume a discrete-time channel with stationary and ergodic time-varying normalized gain $\sqrt{h_m}$ following Rayleigh distribution, and additive white Gaussian noise (AWGN). Suppose that worker m transmits at a rate of $r_m = \log_2(1 + \gamma_{min})$, in which γ_{min} is some fixed minimum received SNR, the data can be correctly received if the instantaneous received SNR $\gamma_m = \frac{P_m h_m}{N_0 B_m}$ is greater than or equal to γ_{min}, in which P_m is the transmission power of worker m; N_0 is the noise power spectral density and B_m is the corresponding bandwidth. The probability of outage is thus $p_{out} = P(\gamma_m < \gamma_{min})$. Particularly, for Rayleigh fading channel, we have

$$p_{out}(r_m) = 1 - e^{-\frac{(2^{r_m}-1)N_0 B_m}{P_m}}. \tag{6.6}$$

The corresponding communication time and energy consumption are given by

$$T_m^{com} = \frac{s_m}{r_m B_m}, \quad E_m^{com} = \frac{P_m s_m}{r_m B_m}, \tag{6.7}$$

in which s_m is the size of the transmitted data (in bits).[2]

For simplicity, the wireless link between each worker m and the parameter server for each entry of the transmitted gradients is assumed to be a binary symmetric channel with crossover probability $p_{out}(r_m)$.[3]

[1] Note that given a fixed transmission rate for the parameter server, the downlink transmission time is a constant that can be readily integrated to the first and the second constraints of the optimization problems (6.15) and (6.16), respectively, if needed.

[2] Note that in the schemes where full precision gradients are transmitted, each worker is supposed to transmit 32 bits for each element in the gradient vectors. Therefore, Algorithm 11 leads to a 32-fold improvement in communication time and communication energy consumption. In addition, s_m also depends on the machine learning model. For instance, in a softmax regression model for k-class classification tasks, $s_m = d \times k$, in which d is the dimension of the gradients.

[3] In this chapter, it is assumed that for each worker m, $sign(g_m^{(t)})$ is transmitted as a single packet in the uplink

6.2 Performance Analysis of Algorithm 11 over Wireless Networks

Before diving into the details of the system design, we first analyze how wireless communications affect the performance of Algorithm 11. To facilitate the analysis, the following commonly adopted assumption is made.

Assumption 8. *(Smoothness). $\forall w_1, w_2$, we require for some non-negative constant L*

$$F(w_1) \leq F(w_2) + < \nabla F(w_2), w_1 - w_2 > + \frac{L}{2}||w_1 - w_2||_2^2, \tag{6.8}$$

where $< \cdot, \cdot >$ is the standard inner product.

Given the above assumption, the following result can be proved.

Theorem 22. *Suppose that the model parameter at the beginning of t-th communication round is $w^{(t)}$, then by performing one communication round of Algorithm 11, we have*

$$\begin{aligned} \mathbb{E}[F(w^{(t)}) - F(w^{(t+1)})] \geq & -\eta||\nabla F(w^{(t)})||_1 - \frac{L\eta^2 d}{2} \\ & + 2\eta \sum_{i=1}^{d} |\nabla F(w^{(t)})_i| P\Big(\tilde{g}_i^{(t)} = sign(\nabla F(w^{(t)}))_i\Big), \end{aligned} \tag{6.9}$$

in which d is the dimension of the gradients; $\tilde{g}_i^{(t)}$ and $\nabla F(w^{(t)})_i$ are the i-th entry of the aggregated result $\tilde{g}^{(t)}$ and the gradient vector $\nabla F(w^{(t)})$, respectively. $sign(\cdot)_i$ is the i-th entry of the vector after taking the sign operation. The expectation and the probability are over the dynamics of the wireless channels.

Proof. Please see Appendix E.1. □

Remark 26. *Theorem 22 lower bounds the expected improvement of the learning objective during the t-th communication round (i.e., $\mathbb{E}[F(w^{(t)}) - F(w^{(t+1)})]$). Intuitively, the learning performance depends on two quantities: (1) the improvement of the learning objective during each communication round; (2) the number of communication rounds. In particular, when the data are homogeneously distributed across the workers, SIGNSGD converges with a rate of $O(1/\sqrt{T})$ [45], in which T is the total number of communication rounds.*

Note that given $w^{(t)}$, $F(w^{(t)})$ and $\nabla F(w^{(t)})$ are constants. Therefore, maximizing the lower bound of the expected improvement of the learning objective (i.e., the right-hand side of (6.9)) is

and all entries of $sign(g_m^{(t)})$ are incorrectly decoded when an outage happens. The analysis in this study is thus conservative. The extension to the scenarios where partial bits of the packet may be recovered is straightforward, and won't change the fundamental tradeoffs revealed in this study.

equivalent to maximizing the probabilities of correct aggregation $P(\tilde{g}_i^{(t)} = sign(\nabla F(w^{(t)}))_i), 1 \leq i \leq d$. For the ease of discussion, we consider the i-th entry of the gradient and define a series of random variables $\{X_m\}_{m=1}^M$ given by

$$X_m = \begin{cases} 1, & \text{if } sign(\hat{g}_m^{(t)})_i \neq sign(\nabla F(w^{(t)}))_i, \\ 0, & \text{if } sign(\hat{g}_m^{(t)})_i = sign(\nabla F(w^{(t)}))_i. \end{cases} \tag{6.10}$$

X_m can be considered as the outcome of one Bernoulli trial with successful probability $P(X_m = 1)$. Let $Z = \sum_{m=1}^M X_m$, then it can be verified that[4]

$$P\Big(\tilde{g}_i^{(t)} = sign(\nabla F(w^{(t)}))_i\Big) = P\bigg(Z < \frac{M}{2}\bigg). \tag{6.11}$$

In addition, Z follows the Poisson binomial distribution with mean $\mathbb{E}[Z] = \sum_{m=1}^M P(X_m = 1)$. Since Z is non-negative, the Markov's inequality gives

$$P\Big(Z \geq \frac{M}{2}\Big) \leq \frac{2\mathbb{E}[Z]}{M}, \tag{6.12}$$

and therefore

$$P\Big(\tilde{g}_i^{(t)} = sign(\nabla F(w^{(t)}))_i\Big) = 1 - P\Big(Z \geq \frac{M}{2}\Big) \geq \frac{M - 2\mathbb{E}[Z]}{M}. \tag{6.13}$$

Note that $\mathbb{E}[Z]$ and $M - \mathbb{E}[Z]$ are the expected number of workers that share wrong and correct signs, respectively. The lower bound in (6.13) represents the difference between the ratios of workers that share the correct signs and that share the wrong signs.

Remark 27. *Note that the closed form of the expected improvement of the learning objective during the t-th communication round (i.e., $\mathbb{E}[F(w^{(t)}) - F(w^{(t+1)})]$) and the probabilities of correct aggregation $P(\tilde{g}_i^{(t)} = sign(\nabla F(w^{(t)}))_i), 1 \leq i \leq d$ are difficult to obtain, especially when the objective function $F(\cdot)$ is unknown. Therefore, the bound derived in (6.13) is used to measure the expected improvement during each communication round. In this sense, in order to optimize the learning performance, we need to: (1) maximize $(M - 2\mathbb{E}[Z])/M$; (2) increase the total number of communication rounds given a fixed total training time (until convergence).*[5]

Given any M, maximizing the right-hand side of (6.13) is equivalent to minimizing $\mathbb{E}[Z] = \sum_{m=1}^M P(X_m = 1)$. In particular, let $p_m^{(t)}$ denote the probability of $sign(g_m^{(t)})_i = sign(\nabla F(w^{(t)}))_i$

[4]Note that the scenario in which $F(w^{(t)})_i = 0$ is not considered in our study for simplicity.

[5]Theoretically, it can be shown that $\mathbb{E}[\frac{1}{T}\sum_{t=1}^T ||\nabla F(w^{(t)})||_1] \leq O(1/\sqrt{T})$. In practice, it is usually not expected that the gradients be reduced to 0. The machine learning algorithms are said to converge when the performance stops improving, which usually takes a finite number of communication rounds.

(i.e., the i-th entry of the local gradient of worker m has the same sign as that of the true gradient $\nabla F(w^{(t)})$), it can be shown that

$$P(X_m = 1) = p_m^{(t)} p_{out}(r_m) + (1 - p_m^{(t)})(1 - p_{out}(r_m)). \tag{6.14}$$

When $p_m^{(t)} > 0.5$,[6] minimizing $P(X_m = 1)$ is equivalent to minimizing $p_{out}(r_m)$.

6.3 Problem Formulation

In this section, the scenario with homogeneous data distribution across the workers is considered. According to our discussion in Section 6.2, in order to optimize the learning performance, it is desired to minimize the outage probabilities of the workers and maximize the number of communication rounds. In this chapter, the implementation of the FL algorithm given a fixed total training time is considered. In this case, the number of communication rounds is inversely proportional to the time duration for each communication round (i.e., $T_m^{cmp} + T_m^{com}$). In addition, considering that the workers (i.e., the mobile devices) have limited batteries, two optimization problems are formulated. Essentially, the first optimization problem addresses the needs of battery-constrained workers while satisfying the requirement for learning performance (which may be dictated by the server), while the second optimization problem is of more interests to the parameter server, whose goal is to optimize the learning performance without consuming excessive energy for the workers.

In the first optimization problem, the energy consumption is minimized given the learning performance constraint (i.e., the outage probabilities of the workers and the time duration for each communication round). It can be seen from (6.6) that given fixed bandwidth B_m and noise power spectral density N_0, the transmission rate r_m and the transmission power P_m determine the outage probability of worker m. Increasing the transmission power P_m and decreasing the transmission rate r_m both decrease the outage probability. However, according to (6.7), a larger P_m and a smaller r_m result in higher communication energy consumption. In addition, given a fixed time duration for each communication round (i.e., $T_m^{cmp} + T_m^{com}$), decreasing r_m increases the communication time T_m^{com} and therefore requires worker m to increasing the CPU frequency f_m such that the local computation time can be reduced. As a result, the local computation energy consumption of worker m also increases. By solving the first optimization problem, each worker m minimizes its energy consumption by selecting appropriate local computation parameter f_m, communication parameters P_m and r_m while satisfying the learning performance

[6]We note that in the scenario with homogeneous data distribution across the workers, the local gradient $g_m^{(t)}$ of worker m can be understood as a noisy estimate of the gradient $\nabla F(w^{(t)})$. Following the assumption in [45] that the noise is symmetric with zero mean, $p_m^{(t)} > 0.5$ always holds. The discussions concerning the heterogeneous data distribution are provided in Section 6.5.

constraint.

In the second optimization problem, the learning performance is optimized given the energy consumption constraint for the workers. Particularly, minimizing the outage probabilities of the workers (and therefore maximizing $(M - 2\mathbb{E}[Z])/M$) during each communication round and maximizing the total number of communication rounds are conflicting. On the one hand, given fixed transmission power P_m and CPU frequency f_m, increasing the time duration for each communication round allows worker m to select a smaller transmission rate r_m, which decreases its outage probability. On the other hand, given a fixed total training time, a larger time duration for each communication round results in fewer communication rounds. That being said, there exists a tradeoff between the total number of communication rounds and the outage probabilities of the workers. By solving the second optimization problem, the tradeoff is appropriately balanced while satisfying the energy consumption constraint for the workers.

Considering that SIGNSGD converges with a rate of $O(1/\sqrt{T})$, the objective of the second optimization problem is to maximize $\sqrt{T}(M - 2\mathbb{E}[Z])/M$, in which $\sqrt{T}$ captures the impact of the number of communication rounds and $(M - 2\mathbb{E}[Z])/M$ captures the improvement of the learning objective at each communication round (c.f. (6.13)).

6.3.1 Energy Minimization Given Learning Performance Constraint

In this subsection, the energy minimization problem given the requirements for the time duration for each communication round and the outage probability of each worker is considered. Given a constraint $p_{out,m}$ on the outage probability and a constraint T_l on the time duration for each communication round, the goal of worker m is to minimize its energy consumption. The corresponding optimization problem is formulated as follows.

$$\begin{aligned} \min_{f_m, r_m, P_m} \ & \frac{\alpha_m}{2} c_m D_m f_m^2 + \frac{P_m s_m}{r_m B_m} \\ \text{s.t.} \ & \frac{c_m D_m}{f_m} + \frac{s_m}{r_m B_m} \leq T_l, \\ & 1 - e^{-\frac{(2^{r_m}-1)N_0 B_m}{P_m}} \leq p_{out,m}, \\ & P_{min,m} \leq P_m \leq P_{max,m}, \\ & f_{min,m} \leq f_m \leq f_{max,m}, \end{aligned} \tag{6.15}$$

The CPU frequency for local computation f_m, the transmission rate r_m and the transmission power P_m are the parameters to be optimized. The feasible regions of CPU frequency and transmission power of worker m are imposed by the second and the third constraints, respectively. Considering that the time duration for each communication round is determined by the slowest worker (the straggler), T_l is set the same for all the workers. Given a fixed total training time, the

time duration requirement imposes a lower bound on the number of communication rounds, while the outage probability requirement imposes a lower bound on $(M-2\mathbb{E}[Z])/M$. Considering that the learning performance improves as the number of communication rounds and $(M-2\mathbb{E}[Z])/M$ increase, T_l and $p_{out,m}$ specify the worst learning performance that the system will achieve in the considered scenario.

6.3.2 Learning Performance Optimization Given Energy Consumption Constraint

In this subsection, the tradeoff between the total number of communication rounds and the outage probabilities of the workers is investigated. According to the discussion in Section 6.2, $\mathbb{E}[Z]=\sum_{m=1}^{M} p_m^{(t)} p_{out}(r_m)+(1-p_m^{(t)})(1-p_{out}(r_m))$, in which $p_m^{(t)}$ is determined by the local dataset of worker m and therefore unknown to the server. To facilitate the discussion, we assume that $p_m^{(t)}=1, \forall m,t$.[7] Given a fixed total training time, since the number of communication rounds is inversely proportional to the time duration of each communication round, the optimization problem is formulated as follows.

$$\begin{aligned} &\max_{T_l, r_m} \frac{M-2\sum_{m=1}^{M} p_{out}(r_m)}{\sqrt{T_l}} \\ \text{s.t.}\quad &\frac{\alpha_m}{2} c_m D_m f_m^2 + \frac{P_m s_m}{r_m B_m} \leq E_m, \forall m, \\ &\max_m \left\{ \frac{c_m D_m}{f_m} + \frac{s_m}{r_m B_m} \right\} \leq T_l, \end{aligned} \tag{6.16}$$

in which the time duration for each communication round T_l and the transmission rate r_m are the parameters to be optimized. E_m is the energy consumption upper limit for worker m. The first constraint captures the energy consumption requirement for each worker m and the second constraint captures the time duration requirement for each communication round.

Furthermore, we assume that the workers transmit with high SNR and therefore we have

$$p_{out}(r_m) \approx \frac{(2^{r_m}-1)N_0 B_m}{P_m}. \tag{6.17}$$

Remark 28. *We note that in practice, it is usually the parameter server that coordinates the whole training process and sets the learning performance requirements (i.e., the time duration and outage probability requirements for each communication round) for the workers. In this sense,*

[7] Note that when all the workers have the same dataset, $g_m^{(t)}=\nabla F(w^{(t)})$ and therefore $p_m^{(t)}=1, \forall m,t$. In the homogeneous data distribution setting, $g_m^{(t)}$ can be considered as a noisy version of $\nabla F(w^{(t)})$. As long as the noise is not too large (e.g., when the local datasets are large enough), this assumption is approximately true. This is verified in our simulation results.

different from (6.15), the optimization problem (6.16) is of more interests to the parameter server. In this case, the primary goal of the parameter server is to optimize the learning performance rather than the energy consumption of the workers (and therefore the parameter server may set a large E_m for each worker m). As a result, the parameter server prefers that $f_m = f_{max,m}$ and $P_m = P_{max,m}$, $\forall m$, such that the communication time and the outage probabilities are minimized. This can be realized by properly setting the outage probability and time duration requirements for each communication round. More specifically, the parameter server solves (6.16) with $P_m = P_{max,m}$ and $f_m = f_{max,m}$, $\forall m$ and sends the corresponding T_l and $p_{out}(r_m)$ to the workers. If worker m selects a smaller f_m, it has to increase its transmission rate r_m to accommodate the time duration requirement. However, a larger r_m results in a higher outage probability, which violates the outage probability requirement. Similar results can be obtained for the transmission power.

6.4 Optimization of System Parameters for Federated Learning

6.4.1 Energy Minimization Given Outage Probability Constraint

We note that the optimization problem (6.15) is not always feasible. In particular, according to the time duration requirement T_l, it is required that $r_m \geq \frac{s_m}{(T_l - \frac{c_m D_m}{f_{max,m}})B_m}$. Combining it with the power constraint and plugging them into (6.6) yields

$$p_{out}(r_m) \geq 1 - e^{-\frac{\left(2^{\frac{s_m}{(T_l - \frac{c_m D_m}{f_{max,m}})B_m}} - 1\right) N_0 B_m}{P_{max,m}}}, \tag{6.18}$$

which may violate the given outage constraint $p_{out,m}$ at least for some worker m. Therefore, two scenarios are considered.

(1) The optimization problem (6.15) is infeasible

In this case, we assume that $P_m = P_{max,m}$, $f_m = f_{max,m}$ and $r_m = \frac{s_m}{(T_l - \frac{c_m D_m}{f_{max,m}})B_m}$.

Remark 29. *We note that T_l and $p_{out}(r_m)$ are the two most important parameters that determine the performance of the FL algorithm. When the optimization problem (6.15) is infeasible, i.e., the outage probability given in (6.18) exceeds $p_{out,m}$ at least for some worker m, the delay requirement and the outage probability requirement cannot be satisfied simultaneously. Since the time duration for each communication round is supposed to be determined by the slowest worker (the straggler), we assume that each worker accommodates the time duration requirement while reducing the outage probability as best it can.*

(2) The optimization problem (6.15) is feasible

For the ease of presentation, we define $r_m^{(1)} = \log_2\Big(-\frac{P_{min,m}\ln(1-p_{out,m})}{N_0 B_m}+1\Big)$, $r_m^{(2)} = \log_2\Big(-\frac{P_{max,m}\ln(1-p_{out,m})}{N_0 B_m}+1\Big)$, and $r_m^{(3)} = \frac{s_m}{B_m(T_l-\frac{c_m D_m}{f_{max,m}})}$.

Lemma 5. *Given any* $\max\{r_m^{(1)}, r_m^{(3)}\} \le r_m \le r_m^{(2)}$*, the optimal transmission power* P_m^* *is given by*

$$P_m^* = -\frac{N_0 B_m(2^{r_m}-1)}{\ln(1-p_{out,m})}. \tag{6.19}$$

Given any $\max\{r_m^{(1)}, r_m^{(3)}\} \le r_m \le r_m^{(2)}$*, the optimal CPU frequency for local computation is given by*

$$f_m^* = \max\left\{\frac{c_m D_m}{T_l - \frac{s_m}{r_m B_m}}, f_{min,m}\right\}. \tag{6.20}$$

Proof. Please see Appendix E.2. □

Remark 30. *Note that* $[r_m^{(3)}, r_m^{(2)}]$ *defines the feasible region of* r_m*. If* $r_m < r_m^{(3)}$*, the time duration requirement cannot be satisfied even with the maximum* f_m*. Similarly, the outage probability requirement cannot be satisfied even with the maximum* P_m *if* $r_m > r_m^{(2)}$*.*

$r_m^{(1)}$ *denotes the minimum transmission rate that worker* m *is supposed to select. For any* $P_{min,m} \le P_m \le P_{max,m}$*, the outage probability requirement is satisfied (i.e.,* $p_{out}(r_m) \le p_{out,m}$*) for any* $r_m \le r_m^{(1)}$*. Given the time duration requirement* T_l*, a larger* r_m *allows worker* m *to select a smaller* f_m*. Considering that the objective function of the optimization problem (6.15) is a decreasing (increasing) function of* r_m *(*f_m*), we have the optimal transmission rate* $r_m^* \ge r_m^{(1)}$*.*

With Lemma 5 at hand, the optimization problem (6.15) can be reformulated as follows.

$$\begin{aligned} \min_{r_m} \quad & \frac{\alpha_m c_m D_m}{2} z_m^2(r_m) - \frac{N_0 s_m(2^{r_m}-1)}{\ln(1-p_{out,m})r_m} \\ \text{s.t.} \quad & \max\{r_m^{(1)}, r_m^{(3)}\} \le r_m \le r_m^{(2)}, \end{aligned} \tag{6.21}$$

in which $z_m(r_m) = \max\{\frac{c_m D_m}{T_l - \frac{s_m}{r_m B_m}}, f_{min,m}\}$.

It can be verified that the objective in (6.21) is convex and therefore, the widely used subgradient methods [139] can be adopted to solve the optimization problem (6.21).

6.4.2 Learning Performance Optimization Given Energy Consumption Constraint

Lemma 6. *In the optimization problem (6.16), given any fixed T_l, the optimal transmission rate of worker m is given by*

$$r_m^* = \max\left\{\frac{P_m s_m}{B_m(E_m - \frac{\alpha_m}{2}c_m D_m f_m^2)}, \frac{s_m f_m}{B_m f_m T_l - B_m c_m D_m}\right\}. \tag{6.22}$$

Proof. Please see Appendix E.3. □

Let $\mathcal{U} = \left\{m|\frac{P_m s_m}{B_m(E_m - \frac{\alpha_m}{2}c_m D_m f_m^2)} \geq \frac{s_m f_m}{B_m f_m T_l - B_m c_m D_m}\right\}$. According to Lemma 6, the workers can be divided into two groups. The optimal transmission rates of the workers in the first group (i.e., $\mathcal{U}$) is limited by their energy consumption upper limit E_m (i.e., further decreasing the transmission rate results in the violation of the energy consumption constraint), while those of the workers in the second group is limited by the time duration for each communication round T_l (i.e., further decreasing the transmission rate results in the violation of the time duration requirement), which is subject to design. Further define the following two functions:

$$g(x) = \frac{2\sum_{m\in\mathcal{U}}\left(2^{\frac{P_m s_m}{B_m(E_m - \frac{\alpha_m}{2}c_m D_m f_m^2)}} - 1\right)N_0 B_m}{P_m\sqrt{x}} + \frac{2\sum_{m\notin\mathcal{U}}\left(2^{\frac{s_m f_m}{B_m f_m x - B_m c_m D_m}} - 1\right)N_0 B_m}{P_m\sqrt{x}}, \tag{6.23}$$

$$h(x) = \frac{M}{\sqrt{x}}. \tag{6.24}$$

Based on Lemma 6, the optimization problem (6.16) can be reformulated as follows.

$$\begin{aligned} &\min_{T_l} \ \ g(T_l) - h(T_l) \\ &\text{s.t.} \ \ T_l \geq \max_m\left\{\frac{c_m D_m}{f_m}\right\}. \end{aligned} \tag{6.25}$$

It can be verified that both $g(x)$ and $h(x)$ are convex functions of x. Therefore, (6.25) is a difference of convex programming problem, which can be solved by the DCA algorithm [140].

6.5 Extension to the Scenario with Heterogeneous Data Distribution across the workers

The discussions in the previous sections consider the scenario with homogeneous data distribution across the workers. It has been shown that SIGNSGD fails to converge when the the data

are heterogeneously distributed across the workers even when the workers can deliver their information without any error [49]. The following example is provided for further illustration.

Example 2. *Suppose that the i-th entry of worker m's gradient is given as follows*[8]

$$\nabla F_m(w^{(t)})_i = \begin{cases} -1, & \text{if } 1 \leq m \leq M-1, \\ M, & \text{if } m = M. \end{cases} \tag{6.26}$$

In this case, we have

$$sign(\nabla F(w^{(t)}))_i = sign\left(\frac{1}{M}\sum_{m=1}^{M} \nabla F_m(w^{(t)})\right)_i = sign\left(\frac{1}{M}\right) = 1 \tag{6.27}$$

It can be easily verified that

$$P(X_m = 1) = P(sign(\hat{g}_m^{(t)}) \neq sign(\nabla F(w^{(t)}))) = \begin{cases} 1 - p_{out}(r_m), & \text{if } 1 \leq m \leq M-1, \\ p_{out}(r_m), & \text{if } m = M. \end{cases} \tag{6.28}$$

Essentially, when $p_{out}(r_m) = 0,\ \forall m,$

$$sign\left(\sum_{m=1}^{M} sign(\hat{g}_m^{(t)})\right) = sign\left(\sum_{m=1}^{M} sign(\nabla F_m(w^{(t)}))\right) \neq sign(\nabla F(w^{(t)})), \tag{6.29}$$

which leads to wrong aggregation.

In Example 2, it can be observed that for worker $m \in \{1, 2, \cdots, M-1\}$, a smaller $p_{out}(r_m)$ results in a larger $P(X_m = 1)$ and smaller $(M - 2\mathbb{E}[Z])/M$, in sharp contrast with the homogeneous case. In this case, the parameter server obtains wrong aggregation results even if all the workers deliver their information without any error. In addition, when the data are heterogeneously distributed across the workers, the probability of such scenarios that lead to wrong aggregation is unknown since neither the parameter server nor the workers has knowledge about the global objective function $F(\cdot)$ (and therefore $p_m^{(t)}$'s). As a result, the convergence of Algorithm 11 cannot be guaranteed. Therefore, it is of vital importance to develop an algorithm that can deal with heterogeneous data distribution across the workers. With such consideration, a stochastic sign based algorithm (i.e., Algorithm 12), termed as Stochastic SIGNSGD with

[8]Recall that in the homogeneous data distribution setting, the gradients of the workers are considered noisy versions of $\nabla F(w^{(t)})$. As a result, such a scenario as (6.26) happens with a small probability (i.e., $p_m^{(t)}$ is large, $\forall m$). In the heterogeneous data distribution setting, $p_m^{(t)}$'s depend on the local datasets of the workers and may be very different from those in the homogeneous data distribution setting.

Algorithm 12 Stochastic SIGNSGD with majority vote over wireless networks

1: Input: initial weight: w_0; number of workers: M; learning rate: η.
2: **for** $t = 0, 1, \cdots, T$ **do**
3: Each worker m obtains its gradient $\nabla F_m(w^{(t)})$. Then, it estimates its outage probability $p_{out}(r_m)$ and does the following pre-processing

$$(g_m^{(t)})_i = \begin{cases} -sign(\nabla F_m(w^{(t)}))_i, & \text{with probability } p_m^i, \\ sign(\nabla F_m(w^{(t)}))_i, & \text{with probability } 1 - p_m^i, \end{cases} \tag{6.30}$$

where $p_m^i = \frac{\frac{1}{2} - p_{out}(r_m) - b|\nabla F_m(w^{(t)})_i|}{1 - 2p_{out}(r_m)}$ and b is a parameter subject to design. Particularly, $0 < b < \frac{1-2p_{out}(r_m)}{2|\nabla F_m(w^{(t)})_i|}$ such that $p_m^i \in (0, \frac{1}{2})$.

4: Each worker m transmits $sign(g_m^{(t)})$ to the parameter server over wireless links.
5: The parameter server obtains a noisy estimate (denoted by $\hat{g}_m^{(t)}$) of the transmitted information $sign(g_m^{(t)})$ from each worker m and sends the aggregated result $\tilde{g}^{(t)} = sign\big(\frac{1}{M}\sum_{m=1}^{M} \hat{g}_m^{(t)}\big)$ back to the workers.
6: The workers update their local models

$$w^{(t+1)} = w^{(t)} - \eta \tilde{g}^{(t)}. \tag{6.31}$$

7: **end for**

majority vote, is proposed. In particular, compared to Algorithm 11, there is a pre-processing step (i.e., step 3) in Algorithm 12, in which each worker projects each entry of the locally obtained gradient to -1 and +1 with certain probabilities, respectively. The aforementioned issue is alleviated by the stochasticity of the projection. Taking $M = 3$ in Example 2 as an example, it can be verified that

$$p_m^{(t)} = P(sign(g_m^{(t)})_i = sign(\nabla F(w^{(t)}))_i) = \begin{cases} \frac{\frac{1}{2} - p_{out}(r_m) - b|\nabla F_m(w^{(t)})_i|}{1 - 2p_{out}(r_m)}, & \text{if } 1 \le m \le 2, \\ \frac{\frac{1}{2} - p_{out}(r_m) + b|\nabla F_m(w^{(t)})_i|}{1 - 2p_{out}(r_m)}, & \text{if } m = 3. \end{cases} \tag{6.32}$$

Plugging (6.32) into (6.14) yields

$$P(X_m = 1) = \begin{cases} \frac{1}{2} + b, & \text{if } 1 \le m \le 2, \\ \frac{1}{2} - 3b, & \text{if } m = 3. \end{cases} \tag{6.33}$$

Therefore,

$$
\begin{aligned}
P\left(Z<\frac{3}{2}\right) &= P\left(\sum_{m=1}^{3} X_m = 1\right) + P\left(\sum_{m=1}^{3} X_m = 0\right) \\
&= 2\left(\frac{1}{2}+b\right)\left(\frac{1}{2}-b\right)\left(\frac{1}{2}+3b\right) + \left(\frac{1}{2}-b\right)^2\left(\frac{1}{2}-3b\right) + \left(\frac{1}{2}-b\right)^2\left(\frac{1}{2}+3b\right) \\
&= \frac{1}{2}+\frac{1}{2}b-6b^3.
\end{aligned} \tag{6.34}
$$

It can be verified that when $0<b<\frac{1}{\sqrt{12}}$, $P(Z<3/2)>\frac{1}{2}$. That being said, the probability of correct aggregation (c.f. (6.11)) is strictly larger than $\frac{1}{2}$ when b is small enough, based on which the convergence of Algorithm 12 can be established [141]. For more general scenarios, Lemma 7 in the following can be proved.

Lemma 7. *For a sufficiently small b, when $p_{out}(r_m) \leq \min_i\{\frac{1}{2}-b|\nabla F_m(w^{(t)})_i|\}$, in which $\nabla F_m(w^{(t)})_i$ is the i-th entry of the gradient $\nabla F_m(w^{(t)})$, we have*

$$
P\Big(\tilde{g}_i^{(t)}=1\Big) = \frac{1}{2} + \frac{\binom{M-1}{\lceil\frac{M+1}{2}\rceil-1}}{2^M}\sum_{m=1}^{M} b\nabla F_m(w^{(t)})_i + O\left(\frac{b^2}{2^M}\right). \tag{6.35}
$$

Proof. Please see Appendix E.4. □

Remark 31. *In (6.35), if the second term dominates the third term (i.e., b is sufficiently small), $P\Big(\tilde{g}_i^{(t)}=1\Big)>\frac{1}{2}$ when $\sum_{m=1}^{M} b\nabla F_m(w^{(t)})_i>0$; $P\Big(\tilde{g}_i^{(t)}=1\Big)<\frac{1}{2}$ when $\sum_{m=1}^{M} b\nabla F_m(w^{(t)})_i<0$. That being said, the probability of wrong aggregation is always smaller than 1/2.*

Given Lemma 7, the following theorem can be proved.

Theorem 23. *Suppose Assumption 8 is satisfied and set the learning rate $\eta=\frac{1}{\sqrt{dT}}$, then by running Algorithm 12 for T iterations, we have*

$$
\frac{1}{T}\sum_{t=1}^{T} c||\nabla F(w^{(t)})||_1 \leq \frac{\mathbb{E}[F(w^{(0)})-F(w^{(T+1)})]\sqrt{d}}{\sqrt{T}}, \tag{6.36}
$$

where c is some positive constant.

Proof. Please see Appendix E.5. □

It is worth mentioning that according to Lemma 7 and Theorem 23, the convergence of Algorithm 12 is based on the condition $p_{out}(r_m) \leq \min_i\{\frac{1}{2}-b|\nabla F_m(w^{(t)})_i|\}$. Therefore, in order

to optimize the learning performance, the corresponding constraint needs to be added to the optimization problem (6.16). In the meantime, b should also be optimized. However, since $\nabla F_m(w^{(t)})$'s are unknown to the server, optimizing the learning performance of Algorithm 12 is highly non-trivial and left as our future work.

With such consideration, in this chapter, we mainly consider the energy consumption minimization problem given predetermined b,[9] time duration requirement T_l and the outage probability requirement $p_{out,m}$ for each worker. It can be observed from (6.35) that the probability of correct aggregation is independent of the outage probability $p_{out}(r_m)$. In the meantime, according to (6.15), the feasible region of the energy consumption minimization problem with a smaller $p_{out,m}$ is a subset of that with a larger $p_{out,m}$. As a result, it is optimal to select $p_{out,m} = \min_i\{\frac{1}{2} - b|\nabla F_m(w^{(t)})_i|\}$ at the t-th communication round. In this case, worker m has to obtain $\nabla F_m(w^{(t)})$ before computing $p_{out,m}$. That being said, it has to finish the local computation of the gradients before solving the energy consumption minimization problem. In this sense, we consider a pre-determined local computation CPU frequency f_m for each worker m's energy consumption minimization problem (which can be realized by setting $f_{min,m} = f_{max,m} = f_m$ in (6.15)).

To this end, during each communication round, each worker m first computes its gradient $\nabla F_m(w^{(t)})$ and determines the optimal outage probability requirement $p_{out,m}$ as above. By solving the energy consumption minimization problem (6.15), it obtains the communication parameters P_m and r_m. Then, each worker m estimates its outage probability $p_{out}(r_m)$ (e.g., through (6.6)) and performs the pre-processing step, after which the processed information is transmitted to the parameter server over wireless links.

6.6 Simulation Results

In this section, we examine the performance of the proposed methods through extensive simulations. We implement a soft-max regression model on the well-known MNIST dataset that consists of 10 categories ranging from digit "0" to "9" and a total of 60,000 training samples and 10,000 testing samples. In this case, the dimension of the gradient is $d = 785$ and the size of updates for each worker is $s_m = 7850$ bits for each communication round. It is assumed that there are 10 workers that collaboratively train a global model given a total training time of 50 seconds. For all the workers, we set $\alpha_m = 2 \times 10^{-28}$; $c_m = 20$ cycles/bit; $D_m = 5 \times 10^6$ bits; $f_{min,m} = 0.3$ GHz; $f_{max,m} = 2$ GHz; $P_{min,m} = 0$; $P_{max,m} = 1$ W; $N_0 = 10^{-8}$ W/Hz;

[9] In the implementation of Algorithm 12, it is possible that the predetermined $b \geq \frac{1-2p_{out}(r_m)}{2|\nabla F_m(w^{(t)})_i|}$ for the i-th entry of worker m's gradient such that $p_m^i \leq 0$. In this case, we round p_m^i to 0, and it can be verified that (6.30) is reduced to $(g_m^{(t)})_i = sign(\nabla F_m(w^{(t)}))_i$. That being said, Algorithm 11 is a special case of Algorithm 12 where b is large enough.

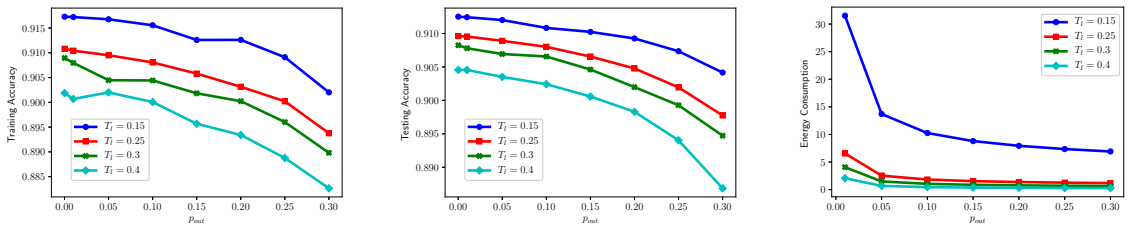

Figure 6.1: The Impact of Outage Probability in the Scenario with Homogeneous Data Distribution across the workers

$B_m = 15$ kHz. In the scenario with homogeneous data distribution across the workers, each worker randomly samples 2000 training samples from the training dataset. In the scenario with heterogeneous data distribution across the workers, the whole training dataset is divided into 10 subsets, each containing the training data for one label. Each worker randomly samples 2000 training samples from one of the subsets.

6.6.1 Energy Minimization Given Learning Performance Constraint: Homogeneous

In this subsection, the impact of the outage probability and time duration for each communication round in (6.15) is examined. We set the same outage probability constraints for all the workers, i.e., $p_{out,m} = p_{out}, \forall m$. The three figures in Fig. 6.1 show the training accuracy, testing accuracy and the average energy consumption for each worker of Algorithm 11 with different p_{out} and T_l, respectively. It can be observed that as p_{out} and T_l increase, the energy consumption decreases. This is because the feasible region of (6.15) corresponding to a smaller p_{out} and T_l is a subset of that of (6.15) corresponding to a larger p_{out} and T_l. On the other hand, both the training accuracy and the testing accuracy decrease as p_{out} and T_l increase. This indicates that there exists a tradeoff between the energy consumption and the learning performance. Given the tradeoff curves, the workers and the parameter server can select suitable p_{out} and T_l to achieve a desired balance between the learning performance and the energy consumption.

6.6.2 Learning Performance Optimization Given Energy Consumption Constraint: Homogeneous

In this subsection, we examine the impact of the transmission power P_m and the time duration for each communication round T_l. The energy consumption upper limit is set as $E_m = 100$ J. Fig. 6.2 shows the performance of Algorithm 11 with different P_m and T_l. Particularly, the training loss is measured by the loss function $F(\cdot)$ corresponding to the soft-max regression model. For the solid curves, the transmission rates r_m's are given by (6.22), while the configurations of the marked points are given by the solution of (6.16). It can be shown that as T_l increases, the

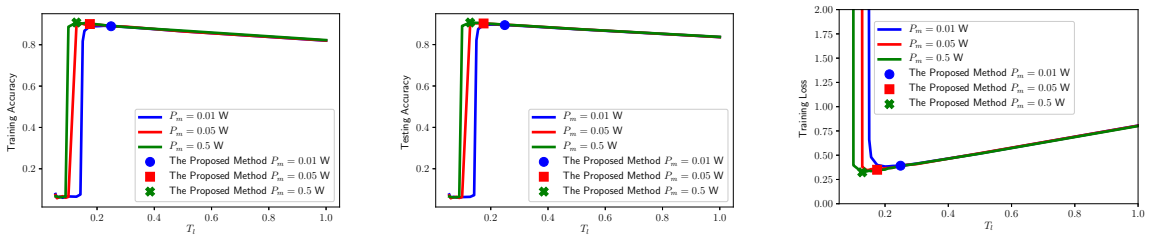

Figure 6.2: The Impact of T_l in the Scenario with Homogeneous Data Distribution Scenario cross Workers

learning performance of Algorithm 11 first increases and then decreases. According to (6.22), when T_l increases, r_m decreases and therefore the outage probability $p_{out}(r_m)$ also decreases. However, in the meantime, as T_l increases, the number of communication rounds decreases given the fixed training time. As a result, when the outage probability has a larger impact on the learning performance, increasing T_l results in better performance. When T_l is larger than a certain critical value, the number of communication rounds plays a more important role and therefore increasing T_l leads to worse performance. In addition, it can be seen that such a critical value decreases as the transmission power increases. Furthermore, Fig. 6.2 shows that the proposed method works close to the optimal operation point for all the examined scenarios, which validates its effectiveness.

6.6.3 Energy Minimization Given Learning Performance Constraint: Heterogeneous

In this subsection, the performance of Algorithm 12 is examined. The CPU frequencies are set as $f_m = 2$ GHz, $\forall m$. The outage probability requirements $p_{out,m}$'s are set according to the discussion in Section 6.5. Fig. 6.3 shows the performance of Algorithm 12 for different b when $T_l = 0.15$. In the "Full Power without Pre-processing" counterpart, we show the performance of Algorithm 12 without the pre-processing step (i.e., step 3) and set $P_m = P_{max,m}$, and $r_m = s_m/(T_l B_m - c_m D_m B_m/f_m)$, i.e., the outage probability $p_{out,m}$ is minimized given that the time duration for each communication round T_l is satisfied. For the "SIGNSGD" baseline, it is assumed that the communication between the workers and the parameter server is perfect (i.e., the outage probabilities are zero). In the considered scenario, the outage probabilities of the workers in the "Full Power without Pre-processing" baseline are small. As a result, "Full Power without Pre-processing" and "SIGNSGD" provide comparable performance. It can be observed that Algorithm 12 outperforms the "Full Power without Pre-processing" and the "SIGNSGD" counterparts for all the examined b's. More specifically, when $b = 0.01$, Algorithm 12 gives an improvement of around 27% in testing accuracy. Table 6.1 shows the corresponding average

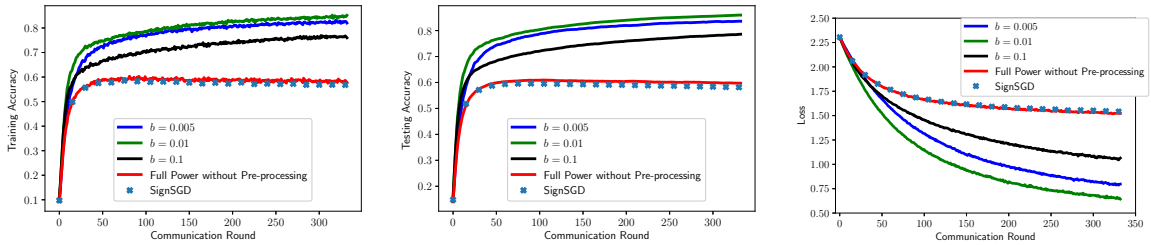

Figure 6.3: The Performance of Algorithm 12 in the Scenario with Heterogeneous Data Distribution across the workers

Table 6.1: Average Energy Consumption of the Workers

b	0.005	0.01	0.1	Full Power without Pre-processing
Energy Consumption (J)	28.37	42.80	46.62	46.62

energy consumption of the workers. It can be observed that when $b = 0.1$, the energy consumption of Algorithm 12 is the same as that of "Full Power without Pre-processing". In this case, the required outage probability $p_{out,m} = \min_i\{\frac{1}{2} - b|\nabla F_m(w^{(t)})_i|\}$ for each worker is small. As a result, as we discussed in Section 6.4.1, the outage probability requirement and the time duration requirement for each communication round cannot be satisfied simultaneously. Therefore, the workers are operating with $P_m = P_{max,m}$. In this case, the only difference between Algorithm 12 and "Full Power without Pre-processing" is the pre-processing step (i.e., step 3) in Algorithm 12. This indicates that the pre-processing step alone gives an improvement of around 20% in testing accuracy as seen in Fig. 6.3. Moreover, Table. 6.1 shows that the average energy consumption increases as b increases. This is because, as b increases, the required outage probability $p_{out,m}$ decreases. As a result, similar to the results in the scenario with homogeneous data distribution across the workers, the average energy consumption of the workers increases. However, it can be observed that different from the scenario with homogeneous data distribution across the workers, increasing b (and therefore decreasing the outage probability) does not necessarily improve the learning performance in the scenario with heterogeneous data distribution across the workers. For instance, Algorithm 12 with $b = 0.01$ performs around 8% better than $b = 0.1$ in testing accuracy. This indicates that compared with Algorithm 11 and "SIGNSGD", Algorithm 12 improves the learning performance and reduces the energy consumption simultaneously by selecting an appropriate b. Furthermore, it can be seen that further decreasing b from 0.01 to 0.005 leads to a degradation of around 3% in testing accuracy while reducing around 30% energy consumption. In practice, a suitable b can be selected to achieve a desired balance between the

learning performance and the energy consumption of the workers.

6.7 Related Works

To improve the communication efficiency of the distributed learning algorithms, various methods have been proposed, including quantization [43–47, 63], sparsification [127, 142, 143] and subsampling [144, 145]. However, most of these works ignore the impact of wireless environments and the resource constraints of the mobile devices, which are of vital importance in the implementation of FL algorithms over real-world wireless networks.

Recently, there have been a number of existing works that study the communication aspects of FL algorithms. In [50, 51], the weighted sum of the training time and the energy consumption is optimized by properly selecting the local computation and the communication parameters. The energy consumption of the communications between the mobile devices and the server is also considered in [52] and the goal is to minimize the weighted sum of the energy consumption and the number of participated mobile devices by mobile device scheduling and effective bandwidth allocation. [53] considers a joint mobile device scheduling and bandwidth allocation problem to minimize the expected FL training time. To further reduce the FL convergence time, [54] incorporates artificial nueral networks (ANNs) to estimate the local FL models of the devices that are not scheduled for transmission. [55] proposes a joint device scheduling and resource block allocation policy for FL under imperfect CSI. [56] considers a cell-free massive MIMO scenario and the training time is minimized by jointly optimizing the local computation and the communication parameters. [57] empirically proposes a learning efficiency metric which is a function of the mini-batch size and the time of each communication round. Resource allocation and the mini-batch size are jointly optimized to maximize the learning efficiency. [58] takes the effect of packet transmission errors into consideration and analyzes its impact on the performance of FL. A joint bandwidth allocation and mobile device selection problem is formulated and solved to minimize a FL loss function that captures the performance of the FL algorithm. However, in these works, the aforementioned effective strategies for improving communication efficiency are not considered.

[59] adopts gradient quantization and proposes a one-bit broadband over-the-air aggregation scheme. The impact of wireless channel hostilities is analyzed. [60–62] propose to combine the quantization, sparsification and error compensation schemes. However,, the energy consumption of the devices as well as the impact of transmission errors are ignored in these works. Moreover, all these works assume CSI at both the transmitter and the receiver sides. In this chapter, we adopt the idea of SIGNSGD with majority vote [45] in the design of the communication system and consider flat-fading channels with receiver only CSI.

6.8 Conclusions

In this chapter, the implementation of the sign based FL algorithms over wireless networks is studied. In particular, considering that the workers have limited batteries, two optimization problems concerning the learning performance and the energy consumption of the workers are formulated and solved for appropriate local processing and communication parameter configuration. Furthermore, since SIGNSGD fails to converge in the scenario with heterogeneous data distribution across the workers, a stochastic sign based algorithm that can deal with data heterogeneity across the workers is proposed and the corresponding energy minimization problem is solved. It is shown that the proposed algorithm improves the learning performance with less energy consumption for the workers. The simulation results demonstrate the effectiveness of the proposed method.

Chapter 7

Summary and Future Work

7.1 Conclusions

While the rapidly developing communication technologies bring collaboration opportunities in various areas, the information exchange in collaboration raises security and privacy concerns. In this dissertation, we investigated the security and privacy problems in three important and emerging areas: collaborative security, crowdsensing, and machine learning.

In Chapter 2, the collaborative IDS configuration problem against rational attackers is tackled through a two-layer SG approach, in which the first layer models the interaction between the attackers and the IDSs while the second layer models the interaction among the IDSs. To solve the two-layer SG, the VCG auction based collaboration scheme is proposed. To further mitigate the communication overhead and complexity issues in VCG auction, a distributed game-theoretic incentive mechanism is also proposed. It is shown that the proposed algorithms can both provide effective collaborative configurations and deliver substantial performance gain as compared to the non-collaborative counterpart. Considering that sharing the security-related information may lead to privacy leakage, the security-privacy tradeoff in collaborative security is investigated in Chapter 3. Three QIF games are formulated to model the interaction among collaborative security entities and the attacker, with each game corresponding to one possible scenario of interest in practice. By solving the games, the optimal strategies for both the attacker and the collaborative entities are obtained.

In Chapter 4, we investigate the AoI minimization problem in the presence of multiple location privacy-aware mobile agents in mobile crowd sensing. The lower bound of the expected payment for the BS is derived as a function of the differential privacy levels for the mobile agents. Then, a payment mechanism that achieves the lower bound is proposed for the BS, which allows it to stimulate the mobile agents to report their locations with its differential privacy levels. Considering that the BS usually has a limited budget, a cost-efficient mobile agent selection

algorithm under budget constraint is also developed.

In Chapter 5, we look into the machine learning area and focus on federated learning which provides privacy guarantees for the mobile devices. In particular, we propose a Stochastic-Sign SGD framework that utilizes two novel gradient compressors and can deal with heterogeneous data distribution. The proposed algorithms are proved to converge in the heterogeneous data distribution scenario. In particular, the proposed differentially private compressor *dp-sign* improves the privacy and the accuracy simultaneously without sacrificing any communication efficiency. Then, we further improve the learning performance of the proposed method by incorporating the error-feedback scheme. In addition, the Byzantine resilience of the proposed algorithms is shown analytically. In Chapter 6, the implementation of the sign based SGD algorithms over wireless networks is studied. Considering that the workers have limited batteries, two optimization problems are formulated and solved for appropriate local processing and communication parameter configuration. The Stochastic-Sign SGD proposed in Chapter 5 is adapted to the wireless settings, which outperforms SIGNSGD in learning performance with less energy consumption for the workers.

7.2 Future Works

Finally, we list below some of the interesting but open topics.

- In the study of collaborative security in Chapter 2 and Chapter 3, the attackers are assumed to be rational. Further consideration of irrational attackers or more intelligent attackers who may explicitly attempt to misguide the security entities remains an interesting direction for future research.

- In Chapter 4, the trajectory of the mobile agents are assumed fixed and known. Designing the trajectory of the mobile agents to maximize their utilities as well as minimize the AoI of the network remains an interesting future direction.

- In Chapter 5 and Chapter 6, each entry of the gradient is represented by 1 bit. In deep learning, however, since the dimension of the gradient vector can be in the order of millions, the communication overhead is still a big concern. Incorporating more aggressive compression methods in FL is another interesting research direction.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.

[2] V. Yegneswaran, P. Barford, and S. Jha, "Global intrusion detection in the DOMINO overlay system," University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2003.

[3] E. Vasilomanolakis, S. Karuppayah, M. Mühlhäuser, and M. Fischer, "Taxonomy and survey of collaborative intrusion detection," *ACM Computing Surveys (CSUR)*, vol. 47, no. 4, pp. 1–33, 2015.

[4] C. J. Fung, "Collaborative intrusion detection networks and insider attacks." *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, vol. 2, no. 1, pp. 63–74, 2011.

[5] Q. Zhu, C. Fung, R. Boutaba, and T. Basar, "A game-theoretical approach to incentive design in collaborative intrusion detection networks," in *2009 International Conference on Game Theory for Networks*. IEEE, 2009, pp. 384–392.

[6] C. J. Fung and R. Boutaba, "Design and management of collaborative intrusion detection networks," in *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*. IEEE, 2013, pp. 955–961.

[7] P. Gross, J. Parekh, and G. Kaiser, "Secure selecticast for collaborative intrusion detection systems," in *Proceedings of the 3rd International Workshop on Distributed Event-Based Systems (DEBS'04)*. IET, 2004.

[8] M. E. Locasto, J. J. Parekh, A. D. Keromytis, and S. J. Stolfo, "Towards collaborative security and p2p intrusion detection," in *Proceedings from the Sixth Annual SMC Information Assurance Workshop*. IEEE, 2005, pp. 333–339.

[9] E. Vasilomanolakis, M. Krügl, C. G. Cordero, M. Mühlhäuser, and M. Fischer, "Skipmon: A locality-aware collaborative intrusion detection system," in *34th International Performance Computing and Communications Conference (IPCCC)*. IEEE, 2015, pp. 1–8.

[10] P. Lincoln, P. A. Porras, and V. Shmatikov, "Privacy-preserving sharing and correlation of security alerts." in *USENIX Security Symposium*, 2004, pp. 239–254.

[11] D. Xu and P. Ning, "Privacy-preserving alert correlation: a concept hierarchy based approach," in *21st Annual Computer Security Applications Conference (ACSAC'05)*. IEEE, 2005, pp. 10–pp.

[12] ——, "A flexible approach to intrusion alert anonymization and correlation," in *2006 Securecomm and Workshops*. IEEE, 2006, pp. 1–10.

[13] H. G. Do and W. K. Ng, "Privacy-preserving approach for sharing and processing intrusion alert data," in *10th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*. IEEE, 2015, pp. 1–6.

[14] J. Cheng, S. H. Wong, H. Yang, and S. Lu, "Smartsiren: virus detection and alert for smartphones," in *Proceedings of the 5th international conference on Mobile systems, applications and services*. ACM, 2007, pp. 258–271.

[15] J. Reed, A. J. Aviv, D. Wagner, A. Haeberlen, B. C. Pierce, and J. M. Smith, "Differential privacy for collaborative security," in *Proceedings of the Third European Workshop on System Security*. ACM, 2010, pp. 1–7.

[16] M. S. Alvim, K. Chatzikokolakis, A. McIver, C. Morgan, C. Palamidessi, and G. Smith, "Axioms for information leakage," in *Computer Security Foundations Symposium (CSF)*. IEEE, 2016, pp. 77–92.

[17] Y. Kawamoto, F. Biondi, and A. Legay, "Hybrid statistical estimation of mutual information for quantifying information flow," in *International Symposium on Formal Methods*. Springer, 2016, pp. 406–425.

[18] J. Heusser and P. Malacaria, "Quantifying information leaks in software," in *Proceedings of the 26th Annual Computer Security Applications Conference*. ACM, 2010, pp. 261–269.

[19] K. Nellore and G. Hancke, "A survey on urban traffic management system using wireless sensor networks," *Sensors*, vol. 16, no. 2, p. 157, 2016.

[20] S. Santini, B. Ostermaier, and R. Adelmann, "On the use of sensor nodes and mobile phones for the assessment of noise pollution levels in urban environments," in *Proc. IEEE International Conference on Networked Sensing Systems (INSS)*, 2009, pp. 1–8.

[21] S. Devarakonda, P. Sevusu, H. Liu, R. Liu, L. Iftode, and B. Nath, "Real-time air quality monitoring through mobile sensing in metropolitan areas," in *Proceedings of the 2nd ACM SIGKDD international workshop on urban computing*, 2013, pp. 1–8.

[22] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *Proc. IEEE INFOCOM*, 2012, pp. 2731–2735.

[23] V. Tripathi, R. Talak, and E. Modiano, "Age optimal information gathering and dissemination on graphs," in *Proc. IEEE Conference on Computer Communications (INFOCOM)*, 2019, pp. 2422–2430.

[24] A. R. Girard, A. S. Howell, and J. K. Hedrick, "Border patrol and surveillance missions using multiple unmanned air vehicles," in *Proc. IEEE Conference on Decision and Control (CDC)*, vol. 1, 2004, pp. 620–625.

[25] J. Liu, X. Wang, B. Bai, and H. Dai, "Age-optimal trajectory planning for uav-assisted data collection," in *Proc. IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2018, pp. 553–558.

[26] J. Wan, J. Liu, Z. Shao, A. Vasilakos, M. Imran, and K. Zhou, "Mobile crowd sensing for traffic prediction in internet of vehicles," *Sensors*, vol. 16, no. 1, p. 88, 2016.

[27] M. Zappatore, A. Longo, and M. A. Bochicchio, "Using mobile crowd sensing for noise monitoring in smart cities," in *Proc. IEEE international multidisciplinary conference on computer and energy science (Splitech)*, 2016, pp. 1–6.

[28] Z. Pan, H. Yu, C. Miao, and C. Leung, "Crowdsensing air quality with camera-enabled mobile devices," in *IAAI Conference*, 2017.

[29] L. Pournajaf, L. Xiong, V. Sunderam, and S. Goryczka, "Spatial task assignment for crowd sensing with cloaked locations," in *Proc. IEEE International Conference on Mobile Data Management*, vol. 1, 2014, pp. 73–82.

[30] L. Pournajaf, D. A. Garcia-Ulloa, L. Xiong, and V. Sunderam, "Participant privacy in mobile crowd sensing task management: A survey of methods and challenges," *ACM Sigmod Record*, vol. 44, no. 4, pp. 23–34, 2016.

[31] L. Wang, D. Zhang, D. Yang, B. Y. Lim, and X. Ma, "Differential location privacy for sparse mobile crowdsensing," in *Proc. IEEE International Conference on Data Mining (ICDM)*, 2016, pp. 1257–1262.

[32] C. Dwork, "Differential privacy," *Encyclopedia of Cryptography and Security*, pp. 338–340, 2011.

[33] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: differential privacy for location-based systems," in *Proc. ACM SIGSAC conference on Computer & communications security*, 2013, pp. 901–914.

[34] L. Wang, D. Yang, X. Han, T. Wang, D. Zhang, and X. Ma, "Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation," in *Proc. International Conference on World Wide Web*, 2017, pp. 627–636.

[35] Y. Xiao and L. Xiong, "Protecting locations with differential privacy under temporal correlations," in *Proc. ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1298–1309.

[36] Y. Xiao, L. Xiong, S. Zhang, and Y. Cao, "Loclok: Location cloaking with differential privacy via hidden markov model," *Proceedings of the VLDB Endowment*, vol. 10, no. 12, pp. 1901–1904, 2017.

[37] V. Primault, A. Boutet, S. B. Mokhtar, and L. Brunie, "The long road to computational location privacy: A survey," *IEEE Communications Surveys & Tutorials*, 2018.

[38] M. Yang, T. Zhu, Y. Xiang, and W. Zhou, "Density-based location preservation for mobile crowdsensing with differential privacy," *IEEE Access*, vol. 6, pp. 14 779–14 789, 2018.

[39] Y. Wang, Z. Cai, X. Tong, Y. Gao, and G. Yin, "Truthful incentive mechanism with location privacy-preserving for mobile crowdsourcing systems," *Computer Networks*, vol. 135, pp. 32–43, 2018.

[40] W. Jin, M. Xiao, M. Li, and L. Guo, "If you do not care about it, sell it: Trading location privacy in mobile crowd sensing," in *Proc. IEEE Conference on Computer Communications (INFOCOM)*, 2019.

[41] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Stand-alone and federated learning under passive and active white-box inference attacks," *arXiv preprint arXiv:1812.00910*, 2018.

[42] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 2, p. 44, 2017.

[43] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Advances in Neural Information Processing Systems*, 2017, pp. 1709–1720.

[44] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li, "TernGrad: Ternary gradients to reduce communication in distributed deep learning," in *Advances in neural information processing systems*, 2017, pp. 1509–1519.

[45] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signSGD: Compressed optimisation for non-convex problems," in *International Conference on Machine Learning*, 2018, pp. 560–569.

[46] J. Wu, W. Huang, J. Huang, and T. Zhang, "Error compensated quantized SGD and its applications to large-scale distributed optimization," in *International Conference on Machine Learning*, 2018, pp. 5325–5333.

[47] N. Agarwal, A. T. Suresh, F. X. X. Yu, S. Kumar, and B. McMahan, "cpSGD: Communication-efficient and differentially-private distributed SGD," in *Advances in Neural Information Processing Systems*, 2018, pp. 7564–7575.

[48] J. Bernstein, J. Zhao, K. Azizzadenesheli, and A. Anandkumar, "signSGD with majority vote is communication efficient and byzantine fault tolerant," in *In Seventh International Conference on Learning Representations (ICLR)*, 2019.

[49] X. Chen, T. Chen, H. Sun, S. Z. Wu, and M. Hong, "Distributed training with heterogeneous data: Bridging median-and mean-based algorithms," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[50] N. H. Tran, W. Bao, A. Zomaya, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *INFOCOM*. IEEE, 2019, pp. 1387–1395.

[51] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Transactions on Wireless Communications*, 2020.

[52] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient radio resource allocation for federated edge learning," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2020, pp. 1–6.

[53] W. Shi, S. Zhou, and Z. Niu, "Device scheduling with fast convergence for wireless federated learning," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[54] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," *arXiv preprint arXiv:2001.07845*, 2020.

[55] M. M. Wadu, S. Samarakoon, and M. Bennis, "Federated learning under channel uncertainty: Joint client scheduling and resource allocation," *arXiv preprint arXiv:2002.00802*, 2020.

[56] T. T. Vu, D. T. Ngo, N. H. Tran, H. Q. Ngo, M. N. Dao, and R. H. Middleton, "Cell-free massive mimo for wireless federated learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6377–6392, 2020.

[57] J. Ren, G. Yu, and G. Ding, "Accelerating dnn training in wireless federated edge learning systems," *IEEE Journal on Selected Areas in Communications*, 2020.

[58] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, 2020.

[59] G. Zhu, Y. Du, D. Gunduz, and K. Huang, "One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis," *arXiv preprint arXiv:2001.05713*, 2020.

[60] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2155–2169, 2020.

[61] ——, "Federated learning over wireless fading channels," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3546–3557, 2020.

[62] M. M. Amiri, D. Gunduz, S. R. Kulkarni, and H. V. Poor, "Update aware device scheduling for federated learning at the wireless edge," *arXiv preprint arXiv:2001.10402*, 2020.

[63] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu, "1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[64] Q. Zhu and T. Başar, "Dynamic policy-based ids configuration," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. IEEE, 2009, pp. 8600–8605.

[65] A. Mas-Colell, M. D. Whinston, J. R. Green *et al.*, *Microeconomic theory*. Oxford university press New York, 1995, vol. 1.

[66] T. H. Ptacek and T. N. Newsham, "Insertion, evasion, and denial of service: Eluding network intrusion detection," SECURE NETWORKS INC CALGARY ALBERTA, Tech. Rep., 1998.

[67] R. Bace and P. Mell, "Nist special publication on intrusion detection systems," BOOZ-ALLEN AND HAMILTON INC MCLEAN VA, Tech. Rep., 2001.

[68] H. Shrobe, "Computational vulnerability analysis for information survivability," *AI Magazine*, vol. 23, no. 4, p. 81, 2002.

[69] J. Hu and M. P. Wellman, "Nash q-learning for general-sum stochastic games," *Journal of machine learning research*, vol. 4, no. Nov, pp. 1039–1069, 2003.

[70] R. A. Miura-Ko, B. Yolken, J. Mitchell, and N. Bambos, "Security decision-making among interdependent organizations," in *2008 21st IEEE Computer Security Foundations Symposium*. IEEE, 2008, pp. 66–80.

[71] T. Başar and G. J. Olsder, *Dynamic noncooperative game theory*. SIAM, Philadelphia, 1998.

[72] R. W. Cottle, J.-S. Pang, and R. E. Stone, *The Linear Complementarity Problem.* SIAM, 1992, vol. 60.

[73] R. De Leone, O. Mangasarian, and T. Shiau, "Multi-sweep asynchronous parallel successive overrelaxation for the nonsymmetric linear complementarity problem," *Annals of Operations Research*, vol. 22, no. 1, pp. 43–54, 1990.

[74] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[75] X. He, H. Dai, P. Ning, and R. Dutta, "Dynamic ids configuration in the presence of intruder type uncertainty," in *2015 IEEE Global Communications Conference (GLOBECOM).* IEEE, 2015, pp. 1–6.

[76] A. Agah, S. K. Das, K. Basu, and M. Asadi, "Intrusion detection in sensor networks: A non-cooperative game approach," in *Third IEEE International Symposium on Network Computing and Applications, 2004.(NCA 2004). Proceedings.* IEEE, 2004, pp. 343–346.

[77] Y. Liu, H. Man, and C. Comaniciu, "A game theoretic approach to efficient mixed strategies for intrusion detection," in *2006 IEEE International Conference on Communications*, vol. 5. IEEE, 2006, pp. 2201–2206.

[78] S. Shen, "A game-theoretic approach for optimizing intrusion detection strategy in WSNs," in *2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC).* IEEE, 2011, pp. 4510–4513.

[79] M. Roesch, "Snort: Lightweight intrusion detection for networks," in *Lisa*, vol. 99, no. 1, 1999, pp. 229–238.

[80] N. Khamphakdee, N. Benjamas, and S. Saiyod, "Improving intrusion detection system based on snort rules for network probe attack detection," in *International Conference on Information and Communication Technology (ICoICT).* IEEE, 2014, pp. 69–74.

[81] C. J. Fung, J. Zhang, I. Aib, and R. Boutaba, "Robust and scalable trust management for collaborative intrusion detection," in *2009 IFIP/IEEE International Symposium on Integrated Network Management*. IEEE, 2009, pp. 33–40.

[82] Q. Zhu, C. Fung, R. Boutaba, and T. Basar, "Guidex: A game-theoretic incentive-based mechanism for intrusion detection networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 11, pp. 2220–2230, 2012.

[83] T. Karygiannis and L. Owens, "Wireless network security," *NIST special publication*, vol. 800, pp. 1–48, 2002.

[84] G. Vache, "Environment characterization and system modeling approach for the quantitative evaluation of security," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2009, pp. 89–102.

[85] K. Horák, Q. Zhu, and B. Bošanskỳ, "Manipulating adversary's belief: A dynamic game approach to deception by design for proactive network security," in *International Conference on Decision and Game Theory for Security*. Springer, 2017, pp. 273–294.

[86] S. A. M'rio, K. Chatzikokolakis, C. Palamidessi, and G. Smith, "Measuring information leakage using generalized gain functions," in *Computer Security Foundations Symposium (CSF)*. IEEE, 2012, pp. 265–279.

[87] G. Smith, "On the foundations of quantitative information flow," in *International Conference on Foundations of Software Science and Computational Structures*. Springer, 2009, pp. 288–302.

[88] J. L. Massey, "Guessing and entropy," in *Proceedings of International Symposium on Information Theory*. IEEE, 1994, pp. 1–204.

[89] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3–55, 2001.

[90] R. B. Ash, *Information Theory*. Dover Publications Inc., New York, 1990.

[91] M. S. Alvim, K. Chatzikokolakis, Y. Kawamoto, and C. Palamidessi, "Information leakage games," in *8th Conference on Decision and Game Theory for Security*. IEEE, 2017.

[92] L. A. Gordon, M. P. Loeb, and W. Lucyshyn, "Sharing information on computer systems security: An economic analysis," *Journal of Accounting and Public Policy*, vol. 22, no. 6, pp. 461–485, 2003.

[93] D. Liu, Y. Ji, and V. Mookerjee, "Knowledge sharing and investment decisions in information security," *Decision Support Systems*, vol. 52, no. 1, pp. 95–107, 2011.

[94] M. Khouzani, V. Pham, and C. Cid, "Strategic discovery and sharing of vulnerabilities in competitive environments," in *International Conference on Decision and Game Theory for Security*. Springer, 2014, pp. 59–78.

[95] D. K. Tosh, S. Sengupta, S. Mukhopadhyay, C. A. Kamhoua, and K. A. Kwiat, "Game theoretic modeling to enforce security information sharing among firms," in *CSCloud*. IEEE, 2015, pp. 7–12.

[96] C. Kamhoua, A. Martin, D. K. Tosh, K. A. Kwiat, C. Heitzenrater, and S. Sengupta, "Cyber-threats information sharing in cloud computing: A game theoretic approach," in *CSCloud*. IEEE, 2015, pp. 382–389.

[97] Z. Yin, D. Korzhyk, C. Kiekintveld, V. Conitzer, and M. Tambe, "Stackelberg vs. Nash in security games: Interchangeability, equivalence, and uniqueness," in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2010, pp. 1139–1146.

[98] D. Monderer and L. S. Shapley, "Potential games," *Games and economic behavior*, vol. 14, no. 1, pp. 124–143, 1996.

[99] Y. Xu, Q. Wu, J. Wang, L. Shen, and A. Anpalagan, "Opportunistic spectrum access using partially overlapping channels: Graphical game and uncoupled learning," *IEEE Transactions on Communications*, vol. 61, no. 9, pp. 3906–3918, 2013.

[100] E. Gal-Or and A. Ghose, "The economic consequences of sharing security information," in *Economics of information security*. Springer, 2004, pp. 95–104.

[101] R. Jin, X. He, and H. Dai, "On the tradeoff between privacy and utility in collaborative intrusion detection systems-a game theoretical approach," in *Proceedings of the Hot Topics in Science of Security: Symposium and Bootcamp*. ACM, 2017, pp. 45–51.

[102] R. Jin, X. He, H. Dai, R. Dutta, and P. Ning, "Towards privacy-aware collaborative security: A game-theoretic approach," in *Privacy-Aware Computing (PAC)*. IEEE, 2017, pp. 72–83.

[103] N. Alon, Y. Emek, M. Feldman, and M. Tennenholtz, "Adversarial leakage in games," *SIAM Journal on Discrete Mathematics*, vol. 27, no. 1, pp. 363–385, 2013.

[104] S. Farhang and J. Grossklags, "Flipleakage: a game-theoretic approach to protect against stealthy attackers in the presence of information leakage," in *International Conference on Decision and Game Theory for Security*. Springer, 2016, pp. 195–214.

[105] B. Zhou and W. Saad, "Joint status sampling and updating for minimizing age of information in the internet of things," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 7468–7482, 2019.

[106] W. Wang, L. Ying, and J. Zhang, "The value of privacy: Strategic data subjects, incentive mechanisms and fundamental limits," *ACM SIGMETRICS Performance Evaluation Review*, vol. 44, no. 1, pp. 249–260, 2016.

[107] ——, "Buying data from privacy-aware individuals: the effect of negative payments," in *International Conference on Web and Internet Economics*. Springer, 2016, pp. 87–101.

[108] P. S. Mogre, M. Hollick, N. d'Heureuse, H. W. Heckel, T. Krop, and R. Steinmetz, "A graph-based simple mobility model," in *Communication in Distributed Systems-15. ITG/GI Symposium.* VDE, 2007, pp. 1–12.

[109] Y. Wang, X. Wu, and D. Hu, "Using randomized response for differential privacy preserving data collection." in *EDBT/ICDT Workshops*, vol. 1558, 2016.

[110] M. A. Abd-Elmagid and H. S. Dhillon, "Average peak age-of-information minimization in uav-assisted IoT networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 2003–2008, 2018.

[111] Y. Chen, O. Sheffet, and S. Vadhan, "Privacy games," in *International Conference on Web and Internet Economics.* Springer, 2014, pp. 371–385.

[112] L. Yang, M. Zhang, S. He, M. Li, and J. Zhang, "Crowd-empowered privacy-preserving data aggregation for mobile crowdsensing," in *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2018, pp. 151–160.

[113] Z. Wang, J. Li, J. Hu, J. Ren, Z. Li, and Y. Li, "Towards privacy-preserving incentive for mobile crowdsensing under an untrusted platform," in *IEEE Conference on Computer Communications*, 2019, pp. 2053–2061.

[114] Z. Zhang, S. He, J. Chen, and J. Zhang, "Reap: An efficient incentive mechanism for reconciling aggregation accuracy and individual privacy in crowdsensing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 12, pp. 2995–3007, 2018.

[115] H. Jin, L. Su, H. Xiao, and K. Nahrstedt, "Incentive mechanism for privacy-aware data aggregation in mobile crowd sensing systems," *IEEE/ACM Transactions on Networking*, vol. 26, no. 5, pp. 2019–2032, 2018.

[116] J. Lin, D. Yang, M. Li, J. Xu, and G. Xue, "Frameworks for privacy-preserving mobile crowdsensing incentive mechanisms," *IEEE Transactions on Mobile Computing*, vol. 17, no. 8, pp. 1851–1864, 2017.

[117] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[118] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[119] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[120] C. Xie, O. Koyejo, and I. Gupta, "Generalized byzantine-tolerant SGD," *arXiv preprint arXiv:1802.10116*, 2018.

[121] L. Xiang, J. Yang, and B. Li, "Differentially-private deep learning from an optimization perspective," in *IEEE Conference on Computer Communications*. IEEE, 2019, pp. 559–567.

[122] S. P. Karimireddy, Q. Rebjock, S. Stich, and M. Jaggi, "Error feedback fixes signSGD and other gradient compression schemes," in *International Conference on Machine Learning*, 2019, pp. 3252–3261.

[123] S. Zheng, Z. Huang, and J. Kwok, "Communication-efficient distributed blockwise momentum SGD with error-feedback," in *Advances in Neural Information Processing Systems*, 2019, pp. 11 446–11 456.

[124] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 308–318.

[125] H. Tang, S. Gan, C. Zhang, T. Zhang, and J. Liu, "Communication compression for decentralized training," in *Advances in Neural Information Processing Systems*, 2018, pp. 7652–7662.

[126] P. Jiang and G. Agrawal, "A linear speedup analysis of distributed deep learning with sparse and quantized communication," in *Advances in Neural Information Processing Systems*, 2018, pp. 2525–2536.

[127] H. Wang, S. Sievert, S. Liu, Z. Charles, D. Papailiopoulos, and S. Wright, "ATOMO: Communication-efficient learning via atomic sparsification," in *Advances in Neural Information Processing Systems*, 2018, pp. 9850–9861.

[128] D. Carlson, Y. P. Hsieh, E. Collins, L. Carin, and V. Cevher, "Stochastic spectral descent for discrete graphical models," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 2, pp. 296–311, 2015.

[129] S. U. Stich, J. B. Cordonnier, and M. Jaggi, "Sparsified SGD with memory," in *Advances in Neural Information Processing Systems*, 2018, pp. 4447–4458.

[130] D. Alistarh, T. Hoefler, M. Johansson, N. Konstantinov, S. Khirirat, and C. Renggli, "The convergence of sparsified gradient methods," in *Advances in Neural Information Processing Systems*, 2018, pp. 5973–5983.

[131] H. Tang, X. Lian, T. Zhang, and J. Liu, "DoubleSqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression," *Proceedings of the 36th International Conference on Machine Learning-Volume 97*, pp. 6155–6165, 2019.

[132] P. Blanchard, R. Guerraoui, J. Stainer *et al.*, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems*, 2017, pp. 119–129.

[133] D. Alistarh, Z. Allen-Zhu, and J. Li, "Byzantine stochastic gradient descent," in *Advances in Neural Information Processing Systems*, 2018, pp. 4613–4623.

[134] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*, 2018, pp. 5650–5659.

[135] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, "RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 1544–1551.

[136] C. Xie, S. Koyejo, and I. Gupta, "SLSGD: Secure and efficient distributed on-device machine learning," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2019.

[137] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 13, no. 2-3, pp. 203–221, 1996.

[138] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[139] S. Boyd, L. Xiao, and A. Mutapcic, "Subgradient methods," *lecture notes of EE392o, Stanford University, Autumn Quarter*, vol. 2004, pp. 2004–2005, 2003.

[140] P. D. Tao and L. T. H. An, "Convex analysis approach to DC programming: theory, algorithms and applications," *Acta mathematica vietnamica*, vol. 22, no. 1, pp. 289–355, 1997.

[141] R. Jin, Y. Huang, X. He, H. Dai, and T. Wu, "Stochastic-Sign SGD for federated learning with theoretical guarantees," *arXiv preprint arXiv:2002.10940*, 2020.

[142] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Sparse binary compression: Towards distributed deep learning with minimal communication," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[143] ——, "Robust and communication-efficient federated learning from non-iid data," *IEEE transactions on neural networks and learning systems*, 2019.

[144] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[145] S. Caldas, J. Konečny, H. B. McMahan, and A. Talwalkar, "Expanding the reach of federated learning by reducing client resource requirements," *arXiv preprint arXiv:1812.07210*, 2018.

APPENDICES

Appendix A

A.1 Proof of Theorem 1

Proof. For the ease of presentation, define $X_i^{U_i} \triangleq \sum_{j=1}^{U_i} e_j^i$. Let $\Delta_j = \sum_i g_{ij} - \sum_i g_{ji}$ denote the amount of resource that IDS_j receives from (or gives to) other IDSs, which satisfies $\sum_{j=1}^{n} \Delta_j = 0$. Further define

$$f_j(\Delta_j) = [(1-q_2)k_j - (q_1-q_2)X_j^{U_j}] - [(1-q_2)k_j - (q_1-q_2)X_j^{U_j+\Delta_j}](1-q_2)^{\Delta_j}, \quad \text{(A.1)}$$

Then according to (2.15) and (2.16), the difference between the IDS's rewards before and after resource allocation is given by

$$\sum_{i=1}^{N} E\{R^{IDS_i}|U_i'\} - \sum_{i=1}^{N} E\{R^{IDS_i}|U_i\} = \sum_{j=1}^{N} f_j(\Delta_j)(1-q_2)^{U_j-1}. \quad \text{(A.2)}$$

When $\Delta_j > 0$, it is easy to see that $X_j^{U_j+\Delta_j} \geqslant X_j^{U_j}$ and $(1-q_2)^{\Delta_j} < 1$, thus $f_j(\Delta_j) > 0$. Similarly, when $\Delta_j < 0$, $f_j(\Delta_j) < 0$.

When there exists j such that $\Delta_j \neq 0$, some terms in (A.2) will be positive and the others will be non-positive. Thus, $(A.2) > 0$ (i.e., there is performance improvement) if and only if the sum of the absolute values of the positive terms are larger than that of the negative terms. Further checking the property of function f_j, it can be verified that

$$f_j(l+2) - f_j(l+1) < f_j(l+1) - f_j(l), \quad \text{(A.3)}$$

which indicates that as l increases, the increasing rate of the function f_j is decreasing. Thus, the necessary and sufficient condition for $(A.2) > 0$ can be transformed into

$$\max_{1\leq j\leq N} |f_j(\Delta_j = 1)(1-q_2)^{U_j-1}| > \min_{1\leq j\leq N} |f_j(\Delta_j = -1)(1-q_2)^{U_j-1}|, \quad \text{(A.4)}$$

which is equivalent to (2.19). □

A.2 Proof of Theorem 2

Proof. If the matrix $\boldsymbol{H}^{(j)}$ is diagonally dominant, then it is also P-matrix [70]. As a result, the corresponding LCP has a unique solution for any $\boldsymbol{b}^j$ [72]. Since $\boldsymbol{H}^{(j)}$'s are diagonally dominant for all $j \in \mathcal{N}$, all the associated LCPs have unique solutions, which means the game has a unique NE. □

A.3 Proof of Theorem 4

Proof. According to (2.35), for any λ_i and $\lambda_i^{'}$,

$$\begin{aligned} |D_i^{'}(\lambda_i) - D_i^{'}(\lambda_i^{'})| &= |\frac{\sum_{j\in C_i}[q_1 e_{l_j}^j + q_2 \sum_{l\neq l_j} e_l^j]}{(c_i+\lambda_i)} - \frac{\sum_{j\in C_i}[q_1 e_{l_j}^j + q_2 \sum_{l\neq l_j} e_l^j]}{(c_i+\lambda_i^{'})}| \\ &= |\frac{\sum_{j\in C_i}[q_1 e_{l_j}^j + q_2 \sum_{l\neq l_j} e_l^j](\lambda_i - \lambda_i^{'})}{(c_i+\lambda_i)(c_i+\lambda_i^{'})}|. \end{aligned} \tag{A.5}$$

Since both λ_i and $\lambda_i^{'}$ are bounded from both above and below, there exists a positive constant K such that

$$|D_i^{'}(\lambda_i) - D_i^{'}(\lambda_i^{'})| \leq K|\lambda_i - \lambda_i^{'}|, \tag{A.6}$$

which means that $D_i^{'}(\lambda_i)$ is Lipschitz continuous with Lipschitz constant K.

As a result, when $\beta_i < \frac{min(2,B_1)}{K}$, (2.38) is a contraction mapping of λ_i and hence the gradient based iterative algorithm converges to dual optimal λ_i^*. □

Appendix B

B.1 Proof of Proposition 1

Proof. According to the monotonicity of posterior vulnerability, when the channel matrix C is not useless and V_g is nonlinear, we have $\hat{V}_g(\pi, C) > V_g(\pi)$. For the attacker's secret, π corresponds to its attacking strategy and C corresponds to entity j's observation and obfuscation process (i.e., $Q^j \times R^j$). For entity j, π corresponds to the distribution of its true observation result and C corresponds to the obfuscation process (i.e., R^j). This can also be interpreted as follows: if the channel matrix is not useless (i.e., contains some information), by observing the obfuscated observation of an entity, the adversary gains information about the true observation result and the leader gains information about the true network state which is only determined by the attacker's action. □

B.2 Proof of Proposition 2

Proof. On the one hand, if entity j does not obfuscate its observation before sharing, its secret is shared directly and the vulnerability is maximized in this case. On the other hand, for the attacker, its attacking strategy can be considered as the prior probability distribution π in (3.7) while the observation matrix Q^j and the obfuscation matrix R^j take the roles of the channel matrix C and post-processing matrix R, respectively. Therefore, according to **DPI**, the vulnerability of the attacker's secret decreases due to the obfuscation process.

Let p_1 and p_2 be two possible misreport probabilities for entity j, with $p_1 < p_2 < 0.5$. Then it can be verified that $R_1^j = \begin{bmatrix} 1-p_1 & p_1 \\ p_1 & 1-p_1 \end{bmatrix}$, $R_2^j = \begin{bmatrix} 1-p_2 & p_2 \\ p_2 & 1-p_2 \end{bmatrix}$, and $R_2^j = \frac{1-2p_2}{1-2p_1} R_1^j + (1 - \frac{1-2p_2}{1-2p_1}) \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$.

According to the convexity of posterior vulnerability function and the monotonicity,

$$\begin{aligned} \hat{V}_g(\pi, R_2^j) &\leq \frac{1-2p_2}{1-2p_1} \hat{V}_g(\pi, R_1^j) + (1 - \frac{1-2p_2}{1-2p_1}) V_g(\pi) \\ &< \frac{1-2p_2}{1-2p_1} \hat{V}_g(\pi, R_1^j) + (1 - \frac{1-2p_2}{1-2p_1}) \hat{V}_g(\pi, R_2^j), \end{aligned} \tag{B.1}$$

which further indicates

$$\hat{V}_g(\pi, R_2^j) < \hat{V}_g(\pi, R_1^j). \tag{B.2}$$

Similarly, it can be verified that for any observation matrix Q^j which is not useless, we have

$$\hat{V}_g(\pi, Q^j R_2^j) < \hat{V}_g(\pi, Q^j R_1^j). \tag{B.3}$$

For entity j, (B.2) indicates that the vulnerability of its secret decreases as its misreport probability increases. In addition, by taking the probability distribution of the attacker's secret before the leader receives the obfuscated observation from entity j as the prior probability distribution π, (B.3) indicates that the vulnerability of the attacker's secret decreases when entity j increases its misreport probability. Therefore, increasing the misreport probability of an entity decreases the vulnerabilities of both the attacker's secret and the entity's secret. □

B.3 Proof of Corollary 2

Proof. In the case in which the action sets of the two players contain two elements each, the solution of the minimax problem can be obtained as (3.16) and (3.17) and Theorem 5 ensures that such solution always exists and is the same as the NE. In addition, when (3.18) holds, it can be verified that

$$\frac{\partial U_E}{\partial p_E(R_L)} \begin{cases} > 0, \text{ if } u_E(\pi_H^A, R_H) - u_E(\pi_H^A, R_L) < 0, \\ < 0, \text{ if } u_E(\pi_H^A, R_H) - u_E(\pi_H^A, R_L) > 0. \end{cases}$$

Therefore, the entities have pure strategy given by (3.19). Similarly, it can be verified that when (3.21) holds, the attacker has pure strategy given by (3.22). Given the pure strategy of the entities (attacker), the attacker (entities) can obtain the optimal strategy that maximizes its own utility. □

B.4 Proof of Lemma 4

Proof. Note that if a mixed strategy is played at an NE, all pure strategies in the support of that mixed strategy must yield an equal expected payoff. Otherwise, a player could profitably deviate to playing the specific strategy in the support that would generate a higher payoff. In addition, the supports of strategies $f(p_A)$ and p_A are the same as suggested by the definition of function f, and the attacker's utility function is the same in Game I and Game II. Therefore, p_A is the best response to p_E in Game II if and only if $f(p_A)$ is the best response to p_E in Game I.

Let $U_E^{G_2}(p_A, p_E)$ denote the utility of the entities if (p_A, p_E) is played in Game II, and $U_E^{G_1}(f(p_A), p_E)$ the utility of the entities if $(f(p_A), p_E)$ is played in Game I. It can be shown

that

$$
\begin{aligned}
&U_E^{G_2}(p_A, p_E) - U_E^{G_2}(p_A, p'_E) \\
&= \sum_{\pi^A \in \mathcal{A}} p_A(\pi^A)[p_E(R_L) - p'_E(R_L)][u_E(\pi^A, R_L) - u_E(\pi^A, R_H)] \\
&= \sum_{\pi^A \in \mathcal{A}} \frac{1}{\eta}\hat{p}_A(\pi^A)[p_E(R_L) - p'_E(R_L)][u_A(\pi^A, R_H) - u_A(\pi^A, R_L)] \\
&= \frac{1}{\eta}[U_E^{G_1}(f(p_A), p_E) - U_E^{G_1}(f(p_A), p'_E)].
\end{aligned}
\tag{B.4}
$$

Therefore $U_E^{G_2}(p_A, p_E) - U_E^{G_2}(p_A, p'_E) \geq 0$ is equivalent to $U_E^{G_1}(f(p_A), p_E) - U_E^{G_1}(f(p_A), p'_E) \geq 0$, which means that p_E is a best response to p_A in Game II if and only if p_E is a best response to $f(p_A)$ in Game I, which completes the proof of Lemma 4. □

B.5 Proof of Theorem 6

Proof. Suppose (p_A, p_E) is an NE strategy in Game II, according to Lemma 4, $(f(p_A), p_E)$ is an NE strategy in Game I. Since Game I is a zero-sum game, it is also a minimax strategy. Considering that the utility functions of the attacker in Game I and Game II are the same, p_E is also a minimax strategy in Game II.

Suppose (p_A, p_E) is a minimax strategy in Game II, then it is also a minimax strategy in Game I, which further indicates that it is an NE strategy in Game I. As a result, according to Lemma 4, there exists a strategy $(f^{-1}(p_A), p_E)$ which is the NE strategy in Game II. Therefore, p_E is also an NE strategy for the entities in Game II. □

Appendix C

C.1 Proof of Theorem 10

Proof. let $\hat{p}_i^s(\hat{\boldsymbol{l}}_t^{-i}, \hat{l}_t^i = k) > 0$ denote the probability that mobile agent i is selected for information delivery given the reported locations from all the other mobile agents $\hat{\boldsymbol{l}}_t^{-i} = [\hat{l}_t^1, \cdots, \hat{l}_t^{i-1}, \hat{l}_t^{i+1}, \cdots, \hat{l}_t^N]$ and its own reported location $\hat{l}_t^i = k$; $R_i(\boldsymbol{l}_t, \hat{\boldsymbol{l}}_t, s_t)$ denote the payment of mobile agent i given the true locations of all the mobile agents $\boldsymbol{l}_t = [l_t^1, l_t^2, \cdots, l_t^N]$, their obfuscated locations $\hat{\boldsymbol{l}}_t$, and the set of nodes that determine to transmit their updates s_t. Then, the utility of agent i, at location k, is given by

$$
\begin{aligned}
&U_i(\epsilon_i, k) \\
&= \sum_{s_t} p(s_t) \Bigg[\frac{e^{\epsilon_i}}{N - 1 + e^{\epsilon_i}} \sum_{\boldsymbol{l}_t} \sum_{\hat{\boldsymbol{l}}_t} p(\boldsymbol{l}_t | l_t^i = k) p(\hat{\boldsymbol{l}}_t | \boldsymbol{l}_t, \hat{l}_t^i = k) \hat{p}_i^s(\hat{\boldsymbol{l}}_t^{-i}, \hat{l}_t^i = k) [R_i(\boldsymbol{l}_t, \hat{\boldsymbol{l}}_t, s_t) - \mathbb{1}(k \in s_t) C_i] \\
&+ \frac{1}{N - 1 + e^{\epsilon}} \sum_{j \neq k} \sum_{\boldsymbol{l}_t} \sum_{\hat{\boldsymbol{l}}_t} p(\boldsymbol{l}_t | l_t^i = k) p(\hat{\boldsymbol{l}}_t | \boldsymbol{l}_t, \hat{l}_t^i = j) \hat{p}_i^s(\hat{\boldsymbol{l}}_t^{-i}, \hat{l}_t^i = j) R_i(\boldsymbol{l}_t, \hat{\boldsymbol{l}}_t, s_t) \Bigg] - g_i(\epsilon_\epsilon),
\end{aligned}
\tag{C.1}
$$

in which $p(\boldsymbol{l}_t | l_t^i = k)$ is the probability that the true locations of the mobile agents are $\boldsymbol{l}_t$ given that the true location of agent i is $l_t^i = k$; $p(\hat{\boldsymbol{l}}_t | \boldsymbol{l}_t, \hat{l}_t^i = k)$ is the probability that the reported locations of the mobile agents are $\hat{\boldsymbol{l}}_t$ given that the true locations of the mobile agents are $\boldsymbol{l}_t$ and the reported location of agent i is $\hat{l}_t^i = k$.

Let

$$
\hat{R}_i(k, k, s_t) = \frac{\sum_{\boldsymbol{l}_t} \sum_{\hat{\boldsymbol{l}}_t} p(\boldsymbol{l}_t | l_t^i = k) p(\hat{\boldsymbol{l}}_t | \boldsymbol{l}_t, \hat{l}_t^i = k) \hat{p}_i^s(\hat{\boldsymbol{l}}_t^{-i}, \hat{l}_t^i = k) R_i(\boldsymbol{l}_t, \hat{\boldsymbol{l}}_t, s_t)}{\sum_{\boldsymbol{l}_t} \sum_{\hat{\boldsymbol{l}}_t} p(\boldsymbol{l}_t | l_t^i = k) p(\hat{\boldsymbol{l}}_t | \boldsymbol{l}_t, \hat{l}_t^i = k) \hat{p}_i^s(\hat{\boldsymbol{l}}_t^{-i}, \hat{l}_t^i = k)}, \tag{C.2}
$$

and

$$\hat{R}_i(k,j,s_t) = \frac{\sum_{\boldsymbol{l}_t}\sum_{\hat{\boldsymbol{l}}_t} p(\boldsymbol{l}_t|l_t^i=j)p(\hat{\boldsymbol{l}}_t|\boldsymbol{l}_t,\hat{l}_t^i=j)\hat{p}_i^s(\hat{\boldsymbol{l}}_t^{-i},\hat{l}_t^i=j)R_i(\boldsymbol{l}_t,\hat{\boldsymbol{l}}_t,s_t)}{\sum_{\boldsymbol{l}_t}\sum_{\hat{\boldsymbol{l}}_t} p(\boldsymbol{l}_t|l_t^i=j)p(\hat{\boldsymbol{l}}_t|\boldsymbol{l}_t,\hat{l}_t^i=j)\hat{p}_i^s(\hat{\boldsymbol{l}}_t^{-i},\hat{l}_t^i=j)}, \tag{C.3}$$

we have

$$\begin{aligned}
&U_i(\epsilon_i,k)\\
&=\sum_{s_t} p(s_t)\left[\frac{e^{\epsilon_i}}{N-1+e^{\epsilon_i}}p_i^s(k)[\hat{R}_i(k,k,s_t)-\mathbb{1}(k\in s_t)C_i]+\frac{1}{N-1+e^{\epsilon_i}}\sum_{j\neq k}p_i^s(j)\hat{R}_i(k,j,s_t)\right]\\
&-g_i(\epsilon_i),
\end{aligned} \tag{C.4}$$

in which $p_i^s(k)=\sum_{\boldsymbol{l}_t}\sum_{\hat{\boldsymbol{l}}_t} p(\boldsymbol{l}_t|l_t^i=k)p(\hat{\boldsymbol{l}}_t|\boldsymbol{l}_t,\hat{l}_t^i=k)\hat{p}_i^s(\hat{\boldsymbol{l}}_t^{-i},\hat{l}_t^i=k)$. In addition,

$$\begin{aligned}
&\frac{\partial U_i(\epsilon_i,k)}{\partial\epsilon_i}\\
&=\sum_{s_t} p(s_t)\left[\frac{(N-1)e^{\epsilon_i}p_i^s(k)[\hat{R}_i(k,k,s_t)-\mathbb{1}(k\in s_t)C_i]}{(N-1+e^{\epsilon_i})^2}-\frac{e^{\epsilon_i}}{(N-1+e^{\epsilon_i})^2}\sum_{j\neq k}p_i^s(j)\hat{R}_i(k,j,s_t)\right]\\
&-g_i'(\epsilon_i)=0.
\end{aligned} \tag{C.5}$$

Therefore,

$$\sum_{s_t} p(s_t)\Big[(N-1)p_i^s(k)[\hat{R}_i(k,k,s_t)-\mathbb{1}(k\in s_t)C_i]-\sum_{j\neq k}p_i^s(j)\hat{R}_i(k,j,s_t)\Big]=\frac{g_i'(\epsilon_i)(N-1+e^{\epsilon_i})^2}{e^{\epsilon_i}}. \tag{C.6}$$

The expected payment of the BS for mobile agent i is given by

$$\begin{aligned}
\bar{P}_i(\epsilon_i) &= \sum_{s_t} p(s_t) \sum_{k=1}^{N} p(l_t^i = k) \left[\frac{e^{\epsilon_i} p_i^s(k)[\hat{R}_i(k,k,s_t) - \mathbb{1}(k \in s_t) C_i] + \sum_{j \neq k} p_i^s(k) \hat{R}_i(k,j,s_t)}{N - 1 + e^{\epsilon_i}} \right] \\
&+ \sum_{s_t} p(s_t) \sum_{k=1}^{N} p(l_t^i = k) \frac{e^{\epsilon_i}}{N - 1 + e^{\epsilon}} p_i^s(k) \mathbb{1}(k \in s_t) C_i \\
&\geq \sum_{s_t} p(s_t) \sum_{k=1}^{N} p(l_t^i = k) \left[\frac{1}{N-1} - \frac{1}{N - 1 + e^{\epsilon_i}} \right] \times \\
&\Big[(N-1) p_i^s(k)[\hat{R}_i(k,k,s_t) - \mathbb{1}(k \in s_t) C_i] - \sum_{j \neq k} p^s(j) \hat{R}_i(k,j,s_t) \Big] \\
&+ \sum_{s_t} p(s_t) \sum_{k=1}^{N} p(l_t^i = k) \frac{e^{\epsilon_i}}{N - 1 + e^{\epsilon_i}} p^s(k) \mathbb{1}(k \in s_t) C_i \\
&= \left[\frac{1}{N-1} - \frac{1}{N - 1 + e^{\epsilon_i}} \right] \frac{g_i'(\epsilon_i)(N - 1 + e^{\epsilon_i})^2}{e^{\epsilon_i}} \\
&+ \sum_{s_t} p(s_t) \sum_{k=1}^{N} p(l_t^i = k) \frac{e^{\epsilon_i}}{N - 1 + e^{\epsilon_i}} p_i^s(k) \mathbb{1}(k \in s_t) C,
\end{aligned} \tag{C.7}$$

In particular, the second inequality is due to the fact that $\frac{\sum_{s_t} p(s_t) \sum_{k=1}^{N} p(l_t^i = k) \sum_{j \neq k} p_i^s(j) R(k,j,s_t)}{N-1} > 0$ and the equality holds when $\sum_{s_t} p(s_t) \sum_{k=1}^{N} p(l_t = k) \sum_{j \neq k} p_i^s(j) \hat{R}_i(k,j,s_t) = 0$; the third equality is due to (C.6). □

C.2 Proof of Theorem 11

Proof. Suppose that the true location of the mobile agent is l_t, given the proposed incentive mechanism, the expected payment $\bar{P}(x)$ of the BS is given by

$$\bar{P}(x) = \frac{e^x}{N - 1 + e^x} p^s(l_t) p^{tr}(l_t) \left[\frac{g'(\epsilon)(N - 1 + e^{\epsilon})^2}{p^s(l_t) p^{tr}(l_t)(N-1) e^{\epsilon}} + C \right],$$

where x is the privacy level of the mobile agent and ϵ is the privacy level that has been determined by the BS. The utility of the mobile agent is therefore given by

$$U_a(x) = \frac{e^x}{N - 1 + e^x} \frac{g'(\epsilon)(N - 1 + e^{\epsilon})^2}{(N-1) e^{\epsilon}} - g(x), \tag{C.8}$$

As a result,

$$\frac{\partial U_a(x)}{\partial x} = \frac{e^x}{(N-1+e^x)^2}\frac{g'(\epsilon)(N-1+e^\epsilon)^2}{e^\epsilon} - g'(x). \tag{C.9}$$

It can be verified that if (4.14) holds, $\frac{\partial U_a(x)}{\partial x} > 0$ when $x < \epsilon$ and $\frac{\partial U_a(x)}{\partial x} < 0$ when $x > \epsilon$. Therefore, the utility of the mobile agent is maximize when $x = \epsilon$. □

C.3 Proof of Theorem 12

Proof. Given that $p_i \geq \frac{1}{N}, \forall i \in \mathcal{N}$, we have $\frac{(N-1)p_i}{1-p_j} \geq \frac{1-p_i}{1-p_j}$, $\frac{1-p_j}{1-p_i} \geq \frac{1-p_j}{(N-1)p_i}$ and $\frac{(N-1)p_j}{1-p_i} \geq \frac{1-p_j}{1-p_i}$. Therefore,

$$\epsilon = \ln(N-1) + \ln\Big(\max\Big\{\frac{p_j}{1-p_i}, \forall i \neq j\Big\}\Big). \tag{C.10}$$

For any p_i, the corresponding terms are $\Big\{\frac{p_i}{1-p_j}, \frac{p_j}{1-p_i}, \forall j\Big\}$. We first show that for any i, the optimal p_i, denoted by p_i^*, satisfies

$$\max\Big\{\frac{p_i^*}{1-p_j^*}, \frac{p_j^*}{1-p_i^*}, \forall j\Big\} = \max\Big\{\frac{p_m^*}{1-p_j^*}, \frac{p_j^*}{1-p_m^*}, \forall m \neq j\Big\}. \tag{C.11}$$

If not, i.e., there exists some k such that $\max\Big\{\frac{p_k^*}{1-p_j^*}, \frac{p_j^*}{1-p_k^*}, \forall j\Big\} < \max\Big\{\frac{p_m^*}{1-p_j^*}, \frac{p_j^*}{1-p_m^*}, \forall m \neq j\Big\}$. Observing that $f_1(p_k) = \frac{p_k}{1-p_j}$ and $f_2(p_k) = \frac{p_j}{1-p_k}$ are both increasing functions of p_k. Then there exist some $\delta > 0$, such that

$$\max\Big\{\frac{p_k^*+\delta}{1-p_j^*}, \frac{p_j^*}{1-(p_k^*+\delta)}, \forall j\Big\} = \max\Big\{\frac{p_m^*}{1-p_j^*}, \frac{p_j^*}{1-p_m^*}, \forall m \neq j\Big\}, \tag{C.12}$$

which means that the agent can increase p_k without increasing the privacy loss. On the other hand, increasing p_k results in higher expected payment, and therefore higher utility for the mobile agent. As a result, for any i, we have

$$\max\Big\{\frac{p_i^*}{1-p_j^*}, \frac{p_j^*}{1-p_i^*}, \forall j\Big\} = \max\Big\{\frac{p_m^*}{1-p_j^*}, \frac{p_j^*}{1-p_m^*}, \forall m \neq j\Big\}. \tag{C.13}$$

Without loss of generality, we assume that $p_1^* \geq p_2^* \geq \cdots \geq p_N^*$, then the above equation reduces to

$$\max\Big\{\frac{p_1^*}{1-p_i^*}, \frac{p_i^*}{1-p_1^*}\Big\} = \max\Big\{\frac{p_1^*}{1-p_j^*}, \frac{p_j^*}{1-p_1^*}\Big\}, \forall 2 \leq i \leq N, 2 \leq j \leq N, \tag{C.14}$$

which can be further verified that

$$p_1^* \geq p_2^* = p_2^* = \cdots = p_N^*. \tag{C.15}$$

The privacy level is then given by

$$\epsilon = \ln(N-1) + \ln\Big(\max\Big\{\frac{p_1^*}{1-p_2^*}, \frac{p_2^*}{1-p_1^*}\Big\}\Big). \tag{C.16}$$

The expected utility of the agent is given by

$$\begin{aligned} U_a(\epsilon) = \sum_{s_t} p(s_t) \Bigg[& p(l_t = 1) p_1^* p^s(1) [R(1,1,s_t) - \mathbb{1}(1 \in s_t) C] \\ & + \sum_{i \geq 2} p(l_t = i) p_2^* p^s(i) [R(i,i,s_t) - \mathbb{1}(i \in s_t) C] \Bigg] - g(\epsilon). \end{aligned} \tag{C.17}$$

Given Assumption 3, it can be verified that, given the proposed incentive mechanism, we have $\forall i \in \mathcal{N}$

$$\begin{aligned} & \sum_{s_t} p(s_t) p(l_t = i) p^s(i) [R(i,i,s_t) - \mathbb{1}(i \in s_t) C] \\ & \leq \sum_{s_t} p(s_t) \sum_{j \neq i} p(l_t = j) p^s(j) [R(j,j,s_t) - \mathbb{1}(j \in s_t) C]. \end{aligned} \tag{C.18}$$

With such consideration, it can be shown that $\frac{p_1^*}{1-p_2^*} = \frac{p_2^*}{1-p_1^*} = \max\Big\{\frac{p_1^*}{1-p_2^*}, \frac{p_2^*}{1-p_1^*}\Big\}$. Otherwise, there are two possible scenarios.

Scenario 1: $\frac{p_1}{1-p_2} < \frac{p_2}{1-p_1}$

In this case, it can be verified that $1 - p_1 < p_2 < p_1$. In addition, we have

$$\frac{\partial \frac{p_1}{1-p_2}}{\partial p_1} = \frac{1}{1-p_2} > 0, \frac{\partial \frac{p_1}{1-p_2}}{\partial p_2} = \frac{p_1}{(1-p_2)^2} > 0, \tag{C.19}$$

and

$$\frac{\partial \frac{p_1}{1-p_2}}{\partial p_1} - \frac{\partial \frac{p_1}{1-p_2}}{\partial p_2} = \frac{1-p_1-p_2}{(1-p_2)^2} < 0. \tag{C.20}$$

This means that there exists a $\delta_1 > 0$ such that $\frac{p_1}{1-p_2} < \frac{p_1-\delta_1}{1-(p_2+\delta_1)} \leq \frac{p_2}{1-p_1}$. On the other hand,

$$\frac{\partial \frac{p_2}{1-p_1}}{\partial p_1} = \frac{p_2}{(1-p_1)^2} > 0, \frac{\partial \frac{p_2}{1-p_1}}{\partial p_2} = \frac{1}{1-p_1} > 0, \tag{C.21}$$

and

$$\frac{\partial \frac{p_2}{1-p_1}}{\partial p_1} - \frac{\partial \frac{p_2}{1-p_1}}{\partial p_2} = \frac{p_1 + p_2 - 1}{(1-p_2)^2} > 0. \tag{C.22}$$

This means that there exists a $\delta_2 > 0$ such that $\frac{p_2+\delta_2}{1-(p_1-\delta_2)} \leq \frac{p_2}{1-p_1}$. Let $\delta = \min\{\delta_1, \delta_2\}$ and consider the strategy $p_1' = p_1 - \delta$ and $p_2' = p_2 + \delta$, we have

$$\max\left\{\frac{p_1'}{1-p_2'}, \frac{p_2'}{1-p_1'}\right\} \leq \max\left\{\frac{p_1}{1-p_2}, \frac{p_2}{1-p_1}\right\}. \tag{C.23}$$

On the other hand, according to (C.18), the payment of the mobile agent with strategy $p_1' = p_1 - \delta, p_2' = p_2 + \delta$ is higher than that of p_1, p_2. Therefore, the mobile agent can increase p_2 and decrease p_1 to achieve higher utility.

Scenario 2: $\frac{p_1}{1-p_2} > \frac{p_2}{1-p_1}$

In this case, it can be verified that $1 - p_1 > p_2$. In addition, we have

$$\frac{\partial \frac{p_1}{1-p_2}}{\partial p_1} = \frac{1}{1-p_2} > 0, \frac{\partial \frac{p_1}{1-p_2}}{\partial p_2} = \frac{p_1}{(1-p_2)^2} > 0, \tag{C.24}$$

and

$$\frac{\partial \frac{p_1}{1-p_2}}{\partial p_1} - \frac{\partial \frac{p_1}{1-p_2}}{\partial p_2} = \frac{1 - p_1 - p_2}{(1-p_2)^2} > 0. \tag{C.25}$$

This means that there exists a $\delta_1 > 0$ such that $\frac{p_1-\delta_1}{1-(p_2+\delta_1)} \leq \frac{p_1}{1-p_2}$. On the other hand,

$$\frac{\partial \frac{p_2}{1-p_1}}{\partial p_1} = \frac{p_2}{(1-p_1)^2} > 0, \frac{\partial \frac{p_2}{1-p_1}}{\partial p_2} = \frac{1}{1-p_1} > 0, \tag{C.26}$$

and

$$\frac{\partial \frac{p_2}{1-p_1}}{\partial p_1} - \frac{\partial \frac{p_2}{1-p_1}}{\partial p_2} = \frac{p_1 + p_2 - 1}{(1-p_2)^2} < 0. \tag{C.27}$$

This means that there exists a $\delta_2 > 0$ such that $\frac{p_2}{1-p_1} < \frac{p_2+\delta_2}{1-(p_1-\delta_2)} \leq \frac{p_1}{1-p_2}$.

Let $\delta = \min\{\delta_1, \delta_2\}$ and consider the strategy $p_1' = p_1 - \delta$ and $p_2' = p_2 + \delta$, we have

$$\max\left\{\frac{p_1'}{1-p_2'}, \frac{p_2'}{1-p_1'}\right\} \leq \max\left\{\frac{p_1}{1-p_2}, \frac{p_2}{1-p_1}\right\}. \tag{C.28}$$

According to (C.18), the payment of the mobile agent with strategy $p_1' = p_1 - \delta, p_2' = p_2 + \delta$ is higher than that of p_1, p_2. Therefore, the agent can increase p_2 and decrease p_1 to achieve higher utility.

Given $\frac{p_1^*}{1-p_2^*} = \frac{p_2^*}{1-p_1^*} = \max\left\{\frac{p_1^*}{1-p_2^*}, \frac{p_2^*}{1-p_1^*}\right\}$ at hand, we have either $p_2^* = 1 - p_1^*$ or $p_2^* = p_1^*$. When $p_2^* = 1 - p_1^*$ the privacy loss is a constant, which means that the mobile agent can again

increase p_2 and decrease p_1 to achieve higher utility. Since $p_2^* \leq p_1^*$, the optimal strategy is $p_2^* = p_1^*$. Furthermore, for any $\frac{1}{N} \leq p < 1$, one can always find some $\epsilon \geq 0$ such that $p = \frac{e^\epsilon}{N-1+e^\epsilon}$, which completes the proof. □

Appendix D

D.1 Proofs

D.1.1 Proof of Theorem 13

Proof. Without loss of generality, assume $u_1 \leq u_2 \leq \cdots \leq u_K < 0 \leq u_{K+1} \leq \cdots \leq u_M$ and $\frac{1}{M}\sum_{m=1}^{M} u_m < 0$. Note that similar analysis can be done when $\frac{1}{M}\sum_{i=m}^{M} u_m > 0$. Further define a series of random variables $\{X_m\}_{m=1}^{M}$ given by

$$X_m = \begin{cases} 1, & \text{if } \hat{u}_m \neq sign\left(\frac{1}{M}\sum_{m=1}^{M} u_m\right), \\ 0, & \text{if } \hat{u}_m = sign\left(\frac{1}{M}\sum_{m=1}^{M} u_m\right). \end{cases} \tag{D.1}$$

In particular, X_m can be considered as the outcome of one Bernoulli trial with successful probability $P(X_m = 1)$. Let $Z = \sum_{m=1}^{M} X_m$ and we have

$$P\left(sign\left(\frac{1}{M}\sum_{m=1}^{M}\hat{u}_m\right) \neq sign\left(\frac{1}{M}\sum_{m=1}^{M} u_m\right)\right) = P\left(Z \geq \frac{M}{2}\right). \tag{D.2}$$

Note that according to the definition of *sto-sign*, b is large enough such that $b > \max_m |u_m|$. The probability of $X_m = 1$ is given by

$$P(X_m = 1) = \frac{b + u_m}{2b}. \tag{D.3}$$

Then, Z follows the Poisson binomial distribution with mean and variance given by

$$\begin{aligned}\mu &= \sum_{m=1}^{M} P(X_m = 1) = \frac{M}{2} + \frac{\sum_{m=1}^{M} u_m}{2b}, \\ \sigma^2 &= \sum_{m=1}^{M} \frac{(b-u_m)(b+u_m)}{4b^2}.\end{aligned} \tag{D.4}$$

For any variable $a > 0$, we have

$$\mathbb{E}\Big[e^{aZ}\Big] = \mathbb{E}\Big[e^{a\sum_{m=1}^{M} X_m}\Big] = \mathbb{E}\Bigg[\prod_{m=1}^{M} e^{aX_m}\Bigg] = \prod_{m=1}^{M} \mathbb{E}\Big[e^{aX_m}\Big], \tag{D.5}$$

where the last equality is due to the independence among X_m's. In addition,

$$\mathbb{E}\Big[e^{aX_m}\Big] = P(X_m = 1)e^a + P(X_m = 0) = 1 + P(X_m = 1)(e^a - 1) \leq e^{P(X_m=1)(e^a-1)}, \tag{D.6}$$

where the last inequality is due to the inequality $1 + y \leq e^y$.

Combining (D.5) and (D.6), we have

$$\mathbb{E}\Big[e^{aZ}\Big] = \prod_{m=1}^{M} \mathbb{E}\Big[e^{aX_m}\Big] \leq \prod_{m=1}^{M} e^{P(X_m=1)(e^a-1)} \leq e^{(e^a-1)\mu}. \tag{D.7}$$

Therefore,

$$P\bigg(Z \geq \frac{M}{2}\bigg) = P\Big(e^{aZ} \geq e^{\frac{Ma}{2}}\Big) \leq \frac{\mathbb{E}\Big[e^{aZ}\Big]}{e^{\frac{Ma}{2}}} \leq \frac{e^{(e^a-1)\mu}}{e^{\frac{Ma}{2}}}, \tag{D.8}$$

where we invoke the Markov's inequality.

Since $\frac{\sum_{m=1}^{M} u_m}{2b} < 0$ by our assumption, it can be verified that $\frac{M}{2\mu} > 1$. Let $a = \ln(\frac{M}{2\mu}) > 0$, we have

$$P\bigg(Z \geq \frac{M}{2}\bigg) \leq \frac{e^{\Big(e^{\ln(\frac{M}{2\mu})}-1\Big)\mu}}{e^{\frac{M\ln(\frac{M}{2\mu})}{2}}} = \frac{e^{\frac{M}{2}-\mu}}{(\frac{M}{2\mu})^{\frac{M}{2}}} = \frac{e^{-\frac{\sum_{m=1}^{M} u_m}{2b}}}{(\frac{M}{M+\frac{1}{b}\sum_{m=1}^{M} u_m})^{\frac{M}{2}}}. \tag{D.9}$$

Let $x = \frac{|\sum_{m=1}^{M} u_m|}{bM}$ and it can be verified that $x < 1$ since $b > max_m|u_m|$. Then (D.9) can be reduced to

$$P\bigg(Z \geq \frac{M}{2}\bigg) \leq \Big[\Big(1-x\Big)e^x\Big]^{\frac{M}{2}}. \tag{D.10}$$

□

D.1.2 Proof of Theorem 14

Proof. Without loss of generality, assume $u_1 \leq u_2 \leq \cdots \leq u_K < 0 \leq u_{K+1} \leq \cdots \leq u_M$. According to the definition of *sto-sign*, we have

$$\hat{u}_m = \textit{sto-sign}(u_m, b) = \begin{cases} 1, \text{ with probability } \frac{b+u_m}{2b}, \\ -1, \text{with probability } \frac{b-u_m}{2b}, \end{cases} \tag{D.11}$$

Further define a series of random variables $\{\hat{X}_m\}_{m=1}^M$ given by

$$\hat{X}_m = \begin{cases} 1, & \text{if } \hat{u}_m = 1, \\ 0, & \text{if } \hat{u}_m = -1. \end{cases} \tag{D.12}$$

In particular, $\hat{X}_m$ can be considered as the outcome of one Bernoulli trial with successful probability $P(\hat{X}_m = 1)$. Let $\hat{Z} = \sum_{m=1}^M \hat{X}_m$, then

$$P\left(sign\left(\frac{1}{M}\sum_{m=1}^{M}\hat{u}_m\right) = 1\right) = P\left(\hat{Z} \geq \frac{M}{2}\right) = \sum_{H=\frac{M+1}{2}}^{M} P(\hat{Z} = H). \tag{D.13}$$

In addition,

$$P(\hat{Z} = H) = \frac{\sum_{A\in F_H}\prod_{i\in A}(b+u_i)\prod_{j\in A^c}(b-u_j)}{(2b)^M} = \frac{a_{M,H}b^M + a_{M-1,H}b^{M-1} + \cdots + a_{0,H}b^0}{(2b)^M}, \tag{D.14}$$

in which F_H is the set of all subsets of H integers that can be selected from $\{1, 2, 3, ..., M\}$; $a_{m,H}, \forall 0 \leq m \leq M$ is some constant. It can be easily verified that $a_{M,H} = \binom{M}{H}$.

When b is sufficiently large, $P(\hat{Z} = H)$ is dominated by the first two terms in (D.14). In particular, $\forall m$, we have

$$\begin{aligned} \sum_{A\in F_H}\prod_{i\in A}(b+u_i)\prod_{j\in A^c}(b-u_j) = (b+u_m)&\sum_{A\in F_H, m\in A}\prod_{i\in A/\{m\}}(b+u_i)\prod_{j\in A^c}(b-u_j) \\ +(b-u_m)&\sum_{A\in F_H, m\notin A}\prod_{i\in A}(b+u_i)\prod_{j\in A^c/\{m\}}(b-u_j). \end{aligned} \tag{D.15}$$

As a result, when $\frac{M+1}{2} \leq H \leq M-1$, the u_m related term in $a_{M-1,H}$ is given by

$$\left[\binom{M-1}{H-1} - \binom{M-1}{H}\right]u_m. \tag{D.16}$$

When $H = M$, the u_m related term in $a_{M-1,H}$ is given by

$$\left[\binom{M-1}{H-1}\right] u_m. \tag{D.17}$$

By summing over m, we have

$$a_{M-1,H} = \left[\binom{M-1}{H-1} - \binom{M-1}{H}\right] \sum_{m=1}^{M} u_m, \tag{D.18}$$

and

$$a_{M-1,H} = \left[\binom{M-1}{H-1}\right] \sum_{m=1}^{M} u_m, \tag{D.19}$$

when $\frac{M+1}{2} \leq H \leq M-1$ and $H = M$, respectively.

By summing over H, we have

$$\sum_{H=\frac{M+1}{2}}^{M} a_{M,H} = \sum_{H=\frac{M+1}{2}}^{M} \binom{M}{H} = 2^{M-1}, \tag{D.20}$$

$$\sum_{H=\frac{M+1}{2}}^{M} a_{M-1,H} = \binom{M-1}{\frac{M-1}{2}} \sum_{m=1}^{M} u_m \tag{D.21}$$

As a result,

$$\begin{aligned} P\left(\hat{Z} \geq \frac{M}{2}\right) = \sum_{H=\frac{M+1}{2}}^{M} P(\hat{Z} = H) &= \frac{2^{M-1}b^M + \binom{M-1}{\frac{M-1}{2}} \sum_{m=1}^{M} u_m b^{M-1}}{(2b)^M} + O\left(\frac{1}{b^2}\right) \\ &= \frac{1}{2} + \frac{\binom{M-1}{\frac{M-1}{2}}}{2^M b} \sum_{m=1}^{M} u_m + O\left(\frac{1}{b^2}\right). \end{aligned} \tag{D.22}$$

Therefore, if the second term dominates the third term (i.e., b is sufficiently large), $P\left(\hat{Z} \geq \frac{M}{2}\right) > \frac{1}{2}$ when $\sum_{m=1}^{M} u_m > 0$; $P\left(\hat{Z} \geq \frac{M}{2}\right) < \frac{1}{2}$ when $\sum_{m=1}^{M} u_m < 0$. That being said, the probability of wrong aggregation is always smaller than 1/2. □

D.1.3 Proof of Theorem 15

Proof. According to Assumption 5, we have

$$
\begin{aligned}
&F(w^{(t+1)}) - F(w^{(t)}) \\
&\leq < \nabla F(w^{(t)}), w^{(t+1)} - w^{(t)} > + \frac{L}{2}||w^{(t+1)} - w^{(t)}||^2 \\
&= -\eta < \nabla F(w^{(t)}), sign\left(\frac{1}{M}\sum_{m=1}^{M} sto\text{-}sign(\boldsymbol{g}_m^{(t)})\right) > + \frac{L}{2}\left|\left|\eta sign\left(\frac{1}{M}\sum_{m=1}^{M} sto\text{-}sign(\boldsymbol{g}_m^{(t)})\right)\right|\right|^2 \\
&= -\eta < \nabla F(w^{(t)}), sign\left(\frac{1}{M}\sum_{m=1}^{M} sto\text{-}sign(\boldsymbol{g}_m^{(t)})\right) > + \frac{L\eta^2 d}{2} \\
&= -\eta||\nabla F(w^{(t)})||_1 + \frac{L\eta^2 d}{2} + 2\eta\sum_{i=1}^{d}|\nabla F(w^{(t)})_i|\mathbb{1}_{sign(\frac{1}{M}\sum_{m=1}^{M} sto\text{-}sign(\boldsymbol{g}_m^{(t)})_i) \neq sign(\nabla F(w^{(t)})_i)},
\end{aligned}
\tag{D.23}
$$

where $\nabla F(w^{(t)})_i$ is the i-th entry of the vector $\nabla F(w^{(t)})$ and η is the learning rate. Taking expectation on both sides yeilds

$$
\begin{aligned}
&\mathbb{E}[F(w^{(t+1)}) - F(w^{(t)})] \leq -\eta||\nabla F(w^{(t)})||_1 + \frac{L\eta^2 d}{2} \\
&+ 2\eta\sum_{i=1}^{d}|\nabla F(w^{(t)})_i| P\left(sign\left(\frac{1}{M}\sum_{m=1}^{M} sto\text{-}sign(\boldsymbol{g}_m^{(t)})_i\right) \neq sign(\nabla F(w^{(t)})_i)\right).
\end{aligned}
\tag{D.24}
$$

Let $\Delta(M)$ denote the solution to $\left[\Big(1-x\Big)e^x\right]^{\frac{M}{2}} = \frac{1-c}{2}$. Since $\left[\Big(1-x\Big)e^x\right]$ is a decreasing function of x for $x < 1$, it can be verified that $\left[\Big(1-x\Big)e^x\right]^{\frac{M}{2}} < \frac{1-c}{2}$ when $x > \Delta(M)$ and $\left[\Big(1-x\Big)e^x\right]^{\frac{M}{2}} \geq \frac{1-c}{2}$ otherwise. According to Theorem 13, we have two possible scenarios as follows.

$$
P\left(sign\left(\frac{1}{M}\sum_{m=1}^{M} sto\text{-}sign(\boldsymbol{g}_m^{(t)})_i\right) \neq sign(\nabla F(w^{(t)})_i)\right)
\begin{cases}
\leq \frac{1-c}{2}, \text{if } \frac{|\nabla F(w^{(t)})_i|}{b} > \Delta(M), \\
\leq 1, \text{if } \frac{|\nabla F(w^{(t)})_i|}{b} \leq \Delta(M).
\end{cases}
\tag{D.25}
$$

Plugging (D.25) into (D.24), we can obtain

$$
\begin{aligned}
&\mathbb{E}[F(w^{(t+1)}) - F(w^{(t)})] \\
&\leq -\eta ||\nabla F(w^{(t)})||_1 + \frac{L\eta^2 d}{2} \\
&+ \eta \left[(1-c) \sum_{i=1}^{d} |\nabla F(w^{(t)})_i| \mathbb{1}_{\frac{|\nabla F(w^{(t)})_i|}{b} > \Delta(M)} + (1-c) \sum_{i=1}^{d} |\nabla F(w^{(t)})_i| \mathbb{1}_{\frac{|\nabla F(w^{(t)})_i|}{b} \leq \Delta(M)} \right] \\
&+ 2\eta \sum_{i=1}^{d} |\nabla F(w^{(t)})_i| \mathbb{1}_{\frac{|\nabla F(w^{(t)})_i|}{b} \leq \Delta(M)} \\
&\leq -\eta ||\nabla F(w^{(t)})||_1 + \frac{L\eta^2 d}{2} + \eta(1-c)||\nabla F(w^{(t)})||_1 + 2\eta \sum_{i=1}^{d} |\nabla F(w^{(t)})_i| \mathbb{1}_{\frac{|\nabla F(w^{(t)})_i|}{b} \leq \Delta(M)} \\
&= -\eta c ||\nabla F(w^{(t)})||_1 + \frac{L\eta^2 d}{2} + 2\eta \sum_{i=1}^{d} |\nabla F(w^{(t)})_i| \mathbb{1}_{\frac{|\nabla F(w^{(t)})_i|}{b} \leq \Delta(M)},
\end{aligned}
\tag{D.26}
$$

Adjusting the above inequality and averaging both sides over $t = 1, 2, \cdots, T$, we can obtain

$$
\begin{aligned}
\frac{1}{T} \sum_{t=1}^{T} \eta c ||\nabla F(w^{(t)})||_1 &\leq \frac{\mathbb{E}[F(w^{(0)}) - F(w^{(T+1)})]}{T} + \frac{L\eta^2 d}{2} \\
&+ \frac{2\eta}{T} \sum_{t=1}^{T} \sum_{i=1}^{d} |\nabla F(w^{(t)})_i| \mathbb{1}_{\frac{|\nabla F(w^{(t)})_i|}{b} \leq \Delta(M)} \\
&\leq \frac{\mathbb{E}[F(w^{(0)}) - F(w^{(T+1)})]}{T} + \frac{L\eta^2 d}{2} + 2\eta db\Delta(M).
\end{aligned}
\tag{D.27}
$$

Letting $\eta = \frac{1}{\sqrt{dT}}$ and dividing both sides by η gives

$$
\begin{aligned}
\frac{1}{T} \sum_{t=1}^{T} c ||\nabla F(w^{(t)})||_1 &\leq \frac{\mathbb{E}[F(w^{(0)}) - F(w^{(T+1)})]\sqrt{d}}{\sqrt{T}} + \frac{L\sqrt{d}}{2\sqrt{T}} + 2db\Delta(M) \\
&\leq \frac{(F(w^{(0)}) - F^*)\sqrt{d}}{\sqrt{T}} + \frac{L\sqrt{d}}{2\sqrt{T}} + 2db\Delta(M),
\end{aligned}
\tag{D.28}
$$

which completes the proof. □

D.1.4 Proof of Theorem 16

Proof. We start from the one-dimension scenario and consider any a, b that satisfy $||a-b||_2 \leq \Delta_2$. Without loss of generality, assume that $dp\text{-}sign(a, \epsilon, \delta) = dp\text{-}sign(b, \epsilon, \delta) = 1$. Then we have

$$\begin{aligned} P(dp\text{-}sign(a, \epsilon, \delta) = 1) &= \Phi\left(\frac{a}{\sigma}\right) = \int_{-\infty}^{a} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} dx, \\ P(dp\text{-}sign(b, \epsilon, \delta) = 1) &= \Phi\left(\frac{b}{\sigma}\right) = \int_{-\infty}^{b} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} dx. \end{aligned} \tag{D.29}$$

Furthermore,

$$\frac{P(dp\text{-}sign(a, \epsilon, \delta) = 1)}{P(dp\text{-}sign(b, \epsilon, \delta) = 1)} = \frac{\int_{-\infty}^{a} e^{-\frac{x^2}{2\sigma^2}} dx}{\int_{-\infty}^{b} e^{-\frac{x^2}{2\sigma^2}} dx} = \frac{\int_{0}^{\infty} e^{-\frac{(x-a)^2}{2\sigma^2}} dx}{\int_{0}^{\infty} e^{-\frac{(x-b)^2}{2\sigma^2}} dx}. \tag{D.30}$$

According to Theorem A.1 in [117], given the parameters ϵ, δ and σ, it can be verified that $e^{-\epsilon} \leq \left|\frac{P(dp\text{-}sign(a,\epsilon,\delta)=1)}{P(dp\text{-}sign(b,\epsilon,\delta)=1)}\right| \leq e^{\epsilon}$ with probability at least $1-\delta$.

For the multi-dimension scenario, consider any vector $\boldsymbol{a}$ and $\boldsymbol{b}$ such that $||\boldsymbol{a}-\boldsymbol{b}||_2 \leq \Delta_2$ and $\boldsymbol{v} \in \{-1, 1\}^d$, we have

$$\frac{P(dp\text{-}sign(\boldsymbol{a}, \epsilon, \delta) = \boldsymbol{v})}{P(dp\text{-}sign(\boldsymbol{b}, \epsilon, \delta) = \boldsymbol{v})} = \frac{\int_D e^{-\frac{||\boldsymbol{x}-\boldsymbol{a}||_2^2}{2\sigma^2}} d\boldsymbol{x}}{\int_D e^{-\frac{||\boldsymbol{x}-\boldsymbol{b}||_2^2}{2\sigma^2}} d\boldsymbol{x}}, \tag{D.31}$$

where D is some integral area depending on $\boldsymbol{v}$. Similarly, it can be shown that $e^{-\epsilon} \leq \left|\frac{P(dp\text{-}sign(\boldsymbol{a},\epsilon,\delta)=\boldsymbol{v})}{P(dp\text{-}sign(\boldsymbol{b},\epsilon,\delta)=\boldsymbol{v})}\right| \leq e^{\epsilon}$ with probability at least $1-\delta$. □

D.1.5 Proof of Theorem 17

Proof. Without loss of generality, assume $u_1 \leq u_2 \leq \cdots \leq u_K < 0 \leq u_{K+1} \leq \cdots \leq u_M$ and $\frac{1}{M}\sum_{i=1}^{M} u_i < 0$. Note that similar analysis can be done when $\frac{1}{M}\sum_{i=1}^{M} u_i > 0$. Further define a series of random variables $\{X_i\}_{i=1}^{M}$ given by

$$X_i = \begin{cases} 1, & \text{if } \hat{u}_i \neq sign\Big(\frac{1}{M}\sum_{i=1}^{M} u_i\Big), \\ 0, & \text{if } \hat{u}_i = sign\Big(\frac{1}{M}\sum_{i=1}^{M} u_i\Big). \end{cases} \tag{D.32}$$

In particular, X_i can be considered as the outcome of one Bernoulli trial with successful

probability $P(X_i = 1)$. Let $Z = \sum_{i=1}^{M} X_i$ and we have

$$P\left(sign\left(\frac{1}{M}\sum_{m=1}^{M}\hat{u}_i\right) \neq sign\left(\frac{1}{M}\sum_{m=1}^{M}u_i\right)\right) = P\left(Z \geq \frac{M}{2}\right). \tag{D.33}$$

In addition,

$$P(X_m = 1) = \Phi\left(\frac{u_m}{\sigma}\right). \tag{D.34}$$

Then, Z follows the Poisson binomial distribution with mean and variance given by

$$\begin{aligned} \mu &= \sum_{m=1}^{M} P(X_m = 1) = \sum_{m=1}^{M} \Phi\left(\frac{u_m}{\sigma}\right), \\ \sigma^2 &= \sum_{m=1}^{M} \Phi\left(\frac{u_m}{\sigma}\right)\left(1 - \Phi\left(\frac{u_m}{\sigma}\right)\right). \end{aligned} \tag{D.35}$$

Let n denote a zero-mean Gaussian noise with variance σ, according to the assumption that $u_1 \leq u_2 \leq \cdots \leq u_K < 0 \leq u_{K+1} \leq \cdots \leq u_M$, we have

$$\begin{aligned} \Phi\left(\frac{u_m}{\sigma}\right) &= \frac{1}{2} - P(u_m < n < 0), \quad \forall 1 \leq m \leq K, \\ \Phi\left(\frac{u_m}{\sigma}\right) &= \frac{1}{2} + P(0 < n < u_m), \quad \forall K+1 \leq m \leq M. \end{aligned} \tag{D.36}$$

Therefore,

$$\mu = \sum_{m=1}^{M} \Phi\left(\frac{u_m}{\sigma}\right) = \frac{M}{2} - \left[\sum_{m=1}^{K} P(u_m < n < 0) - \sum_{m=K+1}^{M} P(0 < n < u_m)\right]. \tag{D.37}$$

Note that for any Gaussian noise, $P(a_1 < n < 0) + P(a_2 < n < 0) \geq P(a_1 + a_2 < n < 0)$ for any $a_1 < 0, a_2 < 0$. Therefore, we consider the worst case scenario such that $\sum_{m=1}^{K} P(u_m < n <$

$0) - \sum_{m=K+1}^{M} P(0 < n < u_m)$ is minimized, i.e., $K = 1$. In this case,

$$\begin{aligned}
&\sum_{m=1}^{K} P(u_m < n < 0) - \sum_{m=K+1}^{M} P(0 < n < u_m) \\
&= P\left(u_1 < n \leq -\sum_{m=2}^{M} u_m\right) + P\left(-\sum_{m=2}^{M} u_m < n < 0\right) - \sum_{m=2}^{M} P(0 < n < u_m) \\
&> \left|\sum_{m=1}^{M} u_m\right| \left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{u_1^2}{2\sigma^2}}\right] + P\left(-\sum_{m=2}^{M} u_m < n < 0\right) - \sum_{m=2}^{M} P(0 < n < u_m) \qquad \text{(D.38)} \\
&> \left|\sum_{m=1}^{M} u_m\right| \left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{u_1^2}{2\sigma^2}}\right] - \left|\sum_{m=2}^{M} u_m\right| \frac{1}{\sqrt{2\pi}\sigma} \left[1 - e^{-\frac{(\sum_{m=2}^{M} u_m)^2}{2\sigma^2}}\right] \\
&= \frac{1}{\sqrt{2\pi}\sigma} \left[\left|\sum_{m=1}^{M} u_m\right| e^{-\frac{u_1^2}{2\sigma^2}} + \left|\sum_{m=2}^{M} u_m\right| \left[e^{-\frac{(\sum_{m=2}^{M} u_m)^2}{2\sigma^2}} - 1\right]\right],
\end{aligned}$$

where the first inequality is due to $f(a) > f(u_1)$ for $a \in (u_1, \sum_{m=1}^{M} u_m]$ and the second inequality is due to $f(a) < \frac{1}{\sqrt{2\pi}\sigma}$ for any $a > 0$, where $f(\cdot)$ is the probability density function of the normal distribution.

In particular, as $\sigma \to \infty$, $|\sum_{m=1}^{M} u_m| e^{-\frac{u_1^2}{2\sigma^2}}$ increases and converges to $|\sum_{m=1}^{M} u_m|$ while $|\sum_{m=2}^{M} u_m| \left[e^{-\frac{(\sum_{m=2}^{M} u_m)^2}{2\sigma^2}} - 1\right]$ increases and converges to 0. Therefore, we have

$$\frac{1}{\sqrt{2\pi}\sigma} \left[\left|\sum_{m=1}^{M} u_m\right| e^{-\frac{u_1^2}{2\sigma^2}} + \left|\sum_{m=2}^{M} u_m\right| \left[e^{-\frac{(\sum_{m=2}^{M} u_m)^2}{2\sigma^2}} - 1\right]\right] \xrightarrow{\sigma \to \infty} -\frac{\sum_{m=1}^{M} u_m}{\sqrt{2\pi}\sigma}. \qquad \text{(D.39)}$$

As a result, there exists a σ_0 such that when $\sigma \geq \sigma_0$, we have

$$\mu = \sum_{m=1}^{M} \Phi\left(\frac{u_m}{\sigma}\right) \leq \frac{M}{2} + \frac{\sum_{m=1}^{M} u_m}{2\sigma}. \qquad \text{(D.40)}$$

Following the same analysis as that in the proof of Theorem 13, we can show that

$$P\left(sign\left(\frac{1}{M}\sum_{m=1}^{M} \hat{u}_i\right) \neq sign\left(\frac{1}{M}\sum_{m=1}^{M} u_i\right)\right) < \left[\left(1 - x\right)e^{x}\right]^{\frac{M}{2}}, \qquad \text{(D.41)}$$

where $x = \frac{|\sum_{m=1}^{M} u_m|}{\sigma M}$. □

D.1.6 Proof of Theorem 18

Proof. By replacing M with $M - k_i$ in (D.8), we can obtain

$$P\left(Z \geq \frac{M-k_i}{2}\right) \leq \frac{e^{(e^a-1)\mu}}{e^{\frac{(M-k_i)a}{2}}}, \tag{D.42}$$

where $\mu = \mathbb{E}[Z]$. It is shown in the proof of Theorem 13 and Theorem 16 that there exists a positive constant s_0 such that when $s > s_0$, $\mu \leq \frac{M}{2} - \frac{|\sum_{m=1}^{M} \nabla f_m(w^{(t)})_i|}{2s}$. Let $a = \ln(\frac{M-k_i}{2\mu})$, we have

$$P\left(Z \geq \frac{M-k_i}{2}\right) \leq \frac{e^{\frac{M-k_i}{2}-\mu}}{(\frac{M-k_i}{2\mu})^{\frac{M-k_i}{2}}} = \left(\frac{e^{\frac{M-2\mu-k_i}{M-k_i}}}{(\frac{M-k_i}{2\mu})}\right)^{\frac{M-k_i}{2}} \leq \left[\Big(1-x\Big)e^x\right]^{\frac{M-k_i}{2}}, \tag{D.43}$$

where $x = \frac{\frac{|\sum_{m=1}^{M} \nabla f_m(w^{(t)})_i|}{s} - k_i}{M-k_i}$. In addition, the above inequality requires $\ln(\frac{M-k_i}{2\mu}) > 0$ and therefore $k_i < \frac{|\sum_{m=1}^{M} \nabla f_m(w^{(t)})_i|}{s}$. □

D.1.7 Proof of Theorem 19

Proof. Note that in the proof of Theorem 15, we obtain

$$\begin{aligned}\mathbb{E}[F(w^{t+1}) - F(w^t)] &\leq -\eta||\nabla F(w^t)||_1 + \frac{L\eta^2 d}{2} \\ &+ 2\eta \sum_{i=1}^{d} |\nabla F(w^t)_i| P\left(sign\left(\frac{1}{M}\sum_{m=1}^{M} q(\boldsymbol{g}_m^t)_i\right) \neq sign\left(\frac{1}{M}\sum_{m=1}^{M} \nabla f_m(w^t)_i\right)\right),\end{aligned} \tag{D.44}$$

where $q(\boldsymbol{g}_m^t) = \textit{sto-sign}(\boldsymbol{g}_m^t)$. For the ease of notation, let

$$\begin{aligned} p_{i,1} &= P\left(sign\left(\frac{1}{M}\sum_{m=1}^{M} q(\boldsymbol{g}_m^t)_i\right) \neq sign\left(\frac{1}{M}\sum_{m=1}^{M} (\boldsymbol{g}_m^t)_i\right)\right), \\ p_{i,2} &= P\left(sign\left(\frac{1}{M}\sum_{m=1}^{M} \nabla f_m(w^t)_i\right) \neq sign\left(\frac{1}{M}\sum_{m=1}^{M} (\boldsymbol{g}_m^t)_i\right)\right) < \frac{1}{2}, \\ p_i &= P\left(sign\left(\frac{1}{M}\sum_{m=1}^{M} q(\boldsymbol{g}_m^t)_i\right) \neq sign\left(\frac{1}{M}\sum_{m=1}^{M} \nabla f_m(w^t)_i\right)\right). \end{aligned} \tag{D.45}$$

Then

$$p_i = p_{i,1}(1-p_{i,2}) + p_{i,2}(1-p_{i,1}) = p_{i,1} + p_{i,2} - 2p_{i,1}p_{i,2}. \tag{D.46}$$

We first prove the convergence under the first condition. According to Theorem 14, we have

$p_{i,1} < \frac{1}{2}$. In this case, it can be verified that p_i is an increasing function of both $p_{i,1}$ and $p_{i,2}$ and therefore $p_i < \frac{1}{2} + \frac{1}{2} - \frac{1}{2} = \frac{1}{2}$. Following similar analysis to that in the proof of Theorem 15, it can be shown that Sto-SIGNSGD converges to the (local) optimum with a rate of $O(\frac{\sqrt{d}}{\sqrt{T}})$.

Then, we prove the convergence under the second condition. According to (D.46), it is obvious that $p_i \leq p_{i,1} + p_{i,2}$. Therefore, we have

$$\sum_{i=1}^{d} |\nabla F(w^t)_i| p_i \leq \sum_{i=1}^{d} |\nabla F(w^t)_i| p_{i,1} + \sum_{i=1}^{d} |\nabla F(w^t)_i| p_{i,2}. \tag{D.47}$$

In particular,

$$\begin{aligned}
p_{i,2} &= P\left(sign\left(\frac{1}{M} \sum_{m=1}^{M} \nabla f_m(w^t)_i \right) \neq sign\left(\frac{1}{M} \sum_{m=1}^{M} (\boldsymbol{g}_m^t)_i \right) \right) \\
&\leq P\left(\left| \frac{1}{M} \sum_{m=1}^{M} \nabla f_m(w^t)_i - \frac{1}{M} \sum_{m=1}^{M} (\boldsymbol{g}_m^t)_i \right| \geq \left| \frac{1}{M} \sum_{m=1}^{M} \nabla f_m(w^t)_i \right| \right) \\
&\leq \frac{\mathbb{E}[|\frac{1}{M} \sum_{m=1}^{M} \nabla f_m(w^t)_i - \frac{1}{M} \sum_{m=1}^{M} (\boldsymbol{g}_m^t)_i|]}{|\frac{1}{M} \sum_{m=1}^{M} \nabla f_m(w^t)_i|} \\
&\leq \frac{\sqrt{\mathbb{E}[(\frac{1}{M} \sum_{m=1}^{M} \nabla f_m(w^t)_i - \frac{1}{M} \sum_{m=1}^{M} (\boldsymbol{g}_m^t)_i)^2]}}{|\frac{1}{M} \sum_{m=1}^{M} \nabla f_m(w^t)_i|} \\
&\leq \frac{\sigma_i}{\sqrt{MT} |\nabla F(w^t)_i|}.
\end{aligned} \tag{D.48}$$

As a result, the second term in (D.47) is bounded by $O(\frac{||\bar{\sigma}||_1}{\sqrt{MT}})$. Following the same analysis as that in the proof of Theorem 15, it can be shown that Sto-SIGNSGD converges with a rate of $O(\frac{1}{\sqrt{T}})$. □

D.1.8 Proof of Theorem 20

The proof of Theorem 20 follows the strategy of taking $y^{(t)} = w^{(t)} - \tilde{\eta}\tilde{\boldsymbol{e}}^{(t)}$ such that $y^{(t)}$ is updated in the same way as $w^{(t)}$ in the non error-feedback scenario. A key technical challenge is to bound the norm of the residual error $||\tilde{\boldsymbol{e}}^{(t)}||_2^2$. Utilizing the fact that the output of the compressor $q(\cdot) \in \{-1, 1\}$, we upper bound it by first proving that in this case, the server's compressor is an α-approximate compressor [122] for some $\alpha < 1$. Therefore, before proving Theorem 20, we first prove the following lemmas.

Lemma 8. *Let $y^{(t)} = w^{(t)} - \eta M \tilde{\boldsymbol{e}}^{(t)}$, we have*

$$y^{(t+1)} = y^{(t)} - \eta \sum_{m=1}^{M} dp\text{-}sign(\boldsymbol{g}_m^{(t)}; \epsilon, \delta). \tag{D.49}$$

Proof.

$$\begin{aligned} y^{(t+1)} &= w^{(t+1)} - \eta M \tilde{\boldsymbol{e}}^{(t+1)} \\ &= w^{(t)} - \eta \tilde{\boldsymbol{g}}^{(t)} - \eta M \tilde{\boldsymbol{e}}^{(t+1)} \\ &= w^{(t)} - \eta \left(\sum_{m=1}^{M} dp\text{-}sign(\boldsymbol{g}_m^{(t)}; \epsilon, \delta) + M\tilde{\boldsymbol{e}}^{(t)} - M\tilde{\boldsymbol{e}}^{(t+1)} \right) - \eta M \tilde{\boldsymbol{e}}^{(t+1)} \\ &= w^{(t)} - \eta \sum_{m=1}^{M} dp\text{-}sign(\boldsymbol{g}_m^{(t)}; \epsilon, \delta) - \eta M \tilde{\boldsymbol{e}}^{(t)} \\ &= y^{(t)} - \eta \sum_{m=1}^{M} dp\text{-}sign(\boldsymbol{g}_m^{(t)}; \epsilon, \delta). \end{aligned} \tag{D.50}$$

□

Lemma 9. *There exists a positive constant $\beta > 0$ such that $\mathbb{E}[||\tilde{\boldsymbol{e}}^{(t)}||_2^2] \leq \beta d, \forall t$.*

Proof. We first prove that for the 1-bit compressor $q(\boldsymbol{g}_m^{(t)})$, there exists some constant α such that the following inequality always holds.

$$\left|\left| \frac{1}{M} \sum_{m=1}^{M} q(\boldsymbol{g}_m^{(t)}) + \tilde{\boldsymbol{e}}^{(t)} - \frac{1}{M} sign\left(\frac{1}{M} \sum_{m=1}^{M} q(\boldsymbol{g}_m^{(t)}) + \tilde{\boldsymbol{e}}^{(t)} \right) \right|\right|_2^2 \leq (1-\alpha) \left|\left| \frac{1}{M} \sum_{m=1}^{M} q(\boldsymbol{g}_m^{(t)}) + \tilde{\boldsymbol{e}}^{(t)} \right|\right|_2^2, \tag{D.51}$$

where $\alpha < 1$ is some positive constant.

For the ease of presentation, we let $\boldsymbol{r}_i^{(t)}$ denote the i-th entry of $\frac{1}{M}\sum_{m=1}^{M} q(\boldsymbol{g}_m^{(t)}) + \tilde{\boldsymbol{e}}^{(t)}$. Then, we can rewrite the left hand side of (D.51) as follows,

$$\left|\left| \frac{1}{M} \sum_{m=1}^{M} q(\boldsymbol{g}_m^{(t)}) + \tilde{\boldsymbol{e}}^{(t)} - \frac{1}{M} sign\left(\frac{1}{M} \sum_{m=1}^{M} q(\boldsymbol{g}_m^{(t)}) + \tilde{\boldsymbol{e}}^{(t)} \right) \right|\right|_2^2 = \sum_{i=1}^{d} \left(\boldsymbol{r}_i^{(t)} - \frac{1}{M} sign(\boldsymbol{r}_i^{(t)}) \right)^2. \tag{D.52}$$

In particular, we have

$$\left(\boldsymbol{r}_i^{(t)} - \frac{1}{M} sign(\boldsymbol{r}_i^{(t)}) \right)^2 = \left((\boldsymbol{r}_i^{(t)})^2 + \frac{1}{M^2} - \frac{2|\boldsymbol{r}_i^{(t)}|}{M} \right) = \left[1 - \frac{1}{M(\boldsymbol{r}_i^{(t)})^2} \left(2|\boldsymbol{r}_i^{(t)}| - \frac{1}{M} \right) \right] (\boldsymbol{r}_i^{(t)})^2. \tag{D.53}$$

If $2|\boldsymbol{r}_i^{(t)}| - \frac{1}{M} > 0, \forall i$, then there exist a positive constant α such that

$$\sum_{i=1}^{d} \left(\boldsymbol{r}_i^{(t)} - \frac{1}{M} sign(\boldsymbol{r}_i^{(t)}) \right)^2 \leq \sum_{i=1}^{d} (1-\alpha)(\boldsymbol{r}_i^{(t)})^2 = (1-\alpha) \left|\left| \frac{1}{M} \sum_{m=1}^{M} q(\boldsymbol{g}_m^{(t)}) + \tilde{\boldsymbol{e}}^{(t)} \right|\right|_2^2. \tag{D.54}$$

In order to prove that $2|\boldsymbol{r}_i^{(t)}| - \frac{1}{M} > 0, \forall i$, we first show that $M(\tilde{\boldsymbol{e}}^{(t)})_i$ is an even number for any t

by induction. In particular, according to Assumption 4 and $(\tilde{\boldsymbol{e}}^{(0)})_i = 0$, $M\boldsymbol{r}_i^{(0)} = \sum_{m=1}^{M} q(\boldsymbol{g}_m^{(0)})_i$ is an odd number. Therefore, $M(\tilde{\boldsymbol{e}}^{(1)})_i = \sum_{m=1}^{M} q(\boldsymbol{g}_m^{(0)})_i - sign(\sum_{m=1}^{M} q(\boldsymbol{g}_m^{(t)})_i)$ is an even number. In addition,

$$M(\tilde{\boldsymbol{e}}^{(t+1)})_i = \sum_{m=1}^{M} q(\boldsymbol{g}_m^{(t)})_i + M(\tilde{\boldsymbol{e}}^{(t)})_i - sign\left(\sum_{m=1}^{M} q(\boldsymbol{g}_m^{(t)})_i + M(\tilde{\boldsymbol{e}}^{(t)})_i\right). \tag{D.55}$$

Given that $M(\tilde{\boldsymbol{e}}^{(t)})_i$ is even, we can show that $M(\tilde{\boldsymbol{e}}^{(t+1)})_i$ is even as well. Therefore, $M\boldsymbol{r}_i^{(t)} = \sum_{m=1}^{M} q(\boldsymbol{g}_m^{(t)})_i + M(\tilde{\boldsymbol{e}}^{(t)})_i$ is odd and $2|\boldsymbol{r}_i^{(t)}| \geq \frac{2}{M} > \frac{1}{M}$, $\forall t, i$.

Given (D.51), we can show that

$$\begin{aligned}
\mathbb{E}||\tilde{\boldsymbol{e}}^{(t+1)}||_2^2 &\leq (1-\alpha)\left|\left|\frac{1}{M}\sum_{m=1}^{M} q(\boldsymbol{g}_m^{(t)}) + \tilde{\boldsymbol{e}}^{(t)}\right|\right|_2^2 \\
&\leq (1-\alpha)(1+\rho)\mathbb{E}||\tilde{\boldsymbol{e}}^{(t)}||_2^2 + (1-\alpha)\left(1+\frac{1}{\rho}\right)\mathbb{E}\left|\left|\frac{1}{M}\sum_{m=1}^{M} q(\boldsymbol{g}_m^{(t)})\right|\right|_2^2 \\
&\leq \sum_{j=0}^{t}[(1-\alpha)(1+\rho)]^{t-j}(1-\alpha)\left(1+\frac{1}{\rho}\right)\mathbb{E}\left|\left|\frac{1}{M}\sum_{m=1}^{M} q(\boldsymbol{g}_m^{(t)})\right|\right|_2^2 \\
&\leq \frac{(1-\alpha)\left(1+\frac{1}{\rho}\right)d}{1-(1-\alpha)(1+\rho)},
\end{aligned} \tag{D.56}$$

where we invoke Young's inequality recurrently and ρ can be any positive constant. Therefore, there exists some constant $\beta > 0$ such that $\mathbb{E}[||\tilde{\boldsymbol{e}}^{(t)}||_2^2] \leq \beta d, \forall t$. □

Now, we are ready to prove Theorem 20.

Proof. Let $y^{(t)} = w^{(t)} - \eta M \tilde{\boldsymbol{e}}^{(t)}$, and $\tilde{\eta} = M\eta$, according to Lemma 8, we have

$$
\begin{aligned}
&\mathbb{E}[F(y^{(t+1)}) - F(y^{(t)})] \\
&\leq -\tilde{\eta}\mathbb{E}\Bigg[< \nabla F(y^{(t)}), \frac{1}{M}\sum_{m=1}^{M} dp\text{-}sign(\boldsymbol{g}_m^{(t)}; \epsilon, \delta) > \Bigg] + \frac{L}{2}\mathbb{E}\Bigg[\Bigg|\Bigg|\tilde{\eta}\frac{1}{M}\sum_{m=1}^{M} dp\text{-}sign(\boldsymbol{g}_m^{(t)}; \epsilon, \delta)\Bigg|\Bigg|_2^2\Bigg] \\
&= \tilde{\eta}\mathbb{E}\Bigg[< \nabla F(w^{(t)}) - \nabla F(y^{(t)}), \frac{1}{M}\sum_{m=1}^{M} dp\text{-}sign(\boldsymbol{g}_m^{(t)}; \epsilon, \delta) > \Bigg] \\
&+ \frac{L\tilde{\eta}^2}{2}\mathbb{E}\Bigg[\Bigg|\Bigg|\frac{1}{M}\sum_{m=1}^{M} dp\text{-}sign(\boldsymbol{g}_m^{(t)}; \epsilon, \delta)\Bigg|\Bigg|_2^2\Bigg] \\
&- \tilde{\eta}\mathbb{E}\Bigg[< \nabla F(w^{(t)}), \frac{1}{M}\sum_{m=1}^{M} dp\text{-}sign(\boldsymbol{g}_m^{(t)}; \epsilon, \delta) > \Bigg].
\end{aligned}
\tag{D.57}
$$

We first bound the first term, in particular, we have

$$
\begin{aligned}
&< \nabla F(w^{(t)}) - \nabla F(y^{(t)}), \frac{1}{M}\sum_{m=1}^{M} dp\text{-}sign(\boldsymbol{g}_m^{(t)}; \epsilon, \delta) > \\
&\leq \frac{\tilde{\eta}}{2}||\frac{1}{M}\sum_{m=1}^{M} dp\text{-}sign(\boldsymbol{g}_m^{(t)}; \epsilon, \delta)||_2^2 + \frac{1}{2\tilde{\eta}}||\nabla F(w^{(t)}) - \nabla F(y^{(t)})||_2^2 \\
&\leq \frac{\tilde{\eta}}{2}||\frac{1}{M}\sum_{m=1}^{M} dp\text{-}sign(\boldsymbol{g}_m^{(t)}; \epsilon, \delta)||_2^2 + \frac{L^2}{2\tilde{\eta}}||y^{(t)} - w^{(t)}||_2^2 \\
&= \frac{\tilde{\eta}}{2}||\frac{1}{M}\sum_{m=1}^{M} dp\text{-}sign(\boldsymbol{g}_m^{(t)}; \epsilon, \delta)||_2^2 + \frac{L^2\tilde{\eta}}{2}||\tilde{\boldsymbol{e}}^{(t)}||_2^2 \\
&\leq \frac{\tilde{\eta}}{2}||\frac{1}{M}\sum_{m=1}^{M} dp\text{-}sign(\boldsymbol{g}_m^{(t)}; \epsilon, \delta)||_2^2 + \frac{L^2\tilde{\eta}\beta d}{2},
\end{aligned}
\tag{D.58}
$$

where the second inequality is due to the L-smoothness of F.

Then, we can bound the last term as follows.

$$
\begin{aligned}
&- \mathbb{E}\left[< \nabla F(w^{(t)}), \frac{1}{M} \sum_{m=1}^{M} dp\text{-}sign(\boldsymbol{g}_m^{(t)}; \epsilon, \delta) > \right] \\
&= -\mathbb{E}\left[\sum_{i=1}^{d} \nabla F(w^{(t)})_i \frac{1}{M} \sum_{m=1}^{M} dp\text{-}sign((\boldsymbol{g}_m^{(t)})_i; \epsilon, \delta) \right] \\
&= -\sum_{i=1}^{d} \nabla F(w^{(t)})_i \frac{1}{M} \sum_{m=1}^{M} \left(2\Phi\left(\frac{\nabla f_m(w^{(t)})_i}{\sigma} \right) - 1 \right) \\
&\leq -\sum_{i=1}^{d} |\nabla F(w^{(t)})_i| \frac{|\nabla F(w^{(t)})_i|}{\sigma} \\
&= -\frac{||\nabla F(w^{(t)})||_2^2}{\sigma},
\end{aligned}
\tag{D.59}
$$

where the inequality is due to (D.40) in the proof of Theorem 17.

Plugging (D.58) and (D.59) into (D.57) yields

$$
\begin{aligned}
&\mathbb{E}[F(y^{(t+1)}) - F(y^{(t)})] \\
&\leq \frac{\tilde{\eta}^2 + L\tilde{\eta}^2}{2} \mathbb{E}\left[\left|\left| \frac{1}{M} \sum_{m=1}^{M} dp\text{-}sign(\boldsymbol{g}_m^{(t)}; \epsilon, \delta) \right|\right|_2^2 \right] + \frac{L^2 \tilde{\eta}^2 \beta d}{2} - \frac{\tilde{\eta} ||\nabla F(w^{(t)})||_2^2}{\sigma} \\
&\leq \frac{(\tilde{\eta}^2 + L\tilde{\eta}^2 + L^2\tilde{\eta}^2\beta)d}{2} - \frac{\tilde{\eta} ||\nabla F(w^{(t)})||_2^2}{\sigma}.
\end{aligned}
\tag{D.60}
$$

Rewriting (D.60) and taking average over $t = 0, 1, 2, \cdots, T-1$ on both sides yields

$$
\begin{aligned}
&\frac{1}{T} \sum_{t=0}^{T-1} \frac{||\nabla F(w^{(t)})||_2^2}{\sigma} \\
&\leq \sum_{t=0}^{T-1} \frac{\mathbb{E}[F(y^{(t)}) - F(y^{(t+1)})]}{\tilde{\eta} T} + \frac{(\tilde{\eta} + L\tilde{\eta} + L^2\tilde{\eta}\beta)d}{2}.
\end{aligned}
\tag{D.61}
$$

Taking $\tilde{\eta} = \frac{1}{\sqrt{Td}}$ and $w^{(0)} = y^{(0)}$ yields

$$
\frac{1}{T} \sum_{t=0}^{T-1} \frac{||\nabla F(w^{(t)})||_2^2}{\sigma} \leq \frac{(F(w^{(0)}) - F^*)\sqrt{d}}{\sqrt{T}} + \frac{(1 + L + L^2\beta)\sqrt{d}}{\sqrt{T}}. \tag{D.62}
$$

□

D.1.9 Proof of Theorem 21

Proof. Without loss of generality, assume that the first M workers are normal and the last B are Byzantine. Following a similar procedure to the proof of Theorem 20, we can show that

$$
\begin{aligned}
&\mathbb{E}[F(y^{(t+1)}) - F(y^{(t)})] \\
&\leq -\tilde{\eta}\mathbb{E}\Bigg[< \nabla F(y^{(t)}), \frac{1}{M+B}\Bigg[\sum_{m=1}^{M} dp\text{-}sign(\boldsymbol{g}_m^{(t)}; \epsilon, \delta) + \sum_{j=1}^{B} byzantine\text{-}sign(\boldsymbol{g}_j^{(t)})\Bigg] > \Bigg] \\
&+ \frac{L}{2}\mathbb{E}\Bigg[\Bigg|\Bigg|\tilde{\eta}\frac{1}{M+B}\Bigg[\sum_{m=1}^{M} dp\text{-}sign(\boldsymbol{g}_m^{(t)}; \epsilon, \delta) + \sum_{j=1}^{B} byzantine\text{-}sign(\boldsymbol{g}_j^{(t)})\Bigg]\Bigg|\Bigg|_2^2\Bigg] \\
&= \tilde{\eta}\mathbb{E}\Bigg[< \nabla F(w^{(t)}) - \nabla F(y^{(t)}), \frac{1}{M+B}\Bigg[\sum_{m=1}^{M} dp\text{-}sign(\boldsymbol{g}_m^{(t)}; \epsilon, \delta) + \sum_{j=1}^{B} byzantine\text{-}sign(\boldsymbol{g}_j^{(t)})\Bigg] > \Bigg] \\
&+ \frac{L\tilde{\eta}^2}{2}\mathbb{E}\Bigg[\Bigg|\Bigg|\frac{1}{M+B}\Bigg[\sum_{m=1}^{M} dp\text{-}sign(\boldsymbol{g}_m^{(t)}; \epsilon, \delta) + \sum_{j=1}^{B} byzantine\text{-}sign(\boldsymbol{g}_j^{(t)})\Bigg]\Bigg|\Bigg|_2^2\Bigg] \\
&- \tilde{\eta}\mathbb{E}\Bigg[< \nabla F(w^{(t)}), \frac{1}{M+B}\Bigg[\sum_{m=1}^{M} dp\text{-}sign(\boldsymbol{g}_m^{(t)}; \epsilon, \delta) + \sum_{j=1}^{B} byzantine\text{-}sign(\boldsymbol{g}_j^{(t)})\Bigg] > \Bigg].
\end{aligned}
\tag{D.63}
$$

For the first term, we have

$$
\begin{aligned}
&< \nabla F(w^{(t)}) - \nabla F(y^{(t)}), \frac{1}{M+B}\Bigg[\sum_{m=1}^{M} dp\text{-}sign(\boldsymbol{g}_m^{(t)}; \epsilon, \delta) + \sum_{j=1}^{B} byzantine\text{-}sign(\boldsymbol{g}_j^{(t)})\Bigg] > \\
&\leq \frac{\tilde{\eta}}{2}\Bigg|\Bigg|\frac{1}{M+B}\Bigg[\sum_{m=1}^{M} dp\text{-}sign(\boldsymbol{g}_m^{(t)}; \epsilon, \delta) + \sum_{j=1}^{B} byzantine\text{-}sign(\boldsymbol{g}_j^{(t)})\Bigg]\Bigg|\Bigg|_2^2 \\
&+ \frac{1}{2\tilde{\eta}}||\nabla F(w^{(t)}) - \nabla F(y^{(t)})||^2 \\
&\leq \frac{\tilde{\eta}d}{2} + \frac{L^2}{2\tilde{\eta}}||y^{(t)} - w^{(t)}||^2 \\
&= \frac{\tilde{\eta}d}{2} + \frac{L^2\tilde{\eta}}{2}||\tilde{\boldsymbol{e}}^{(t)}||^2 \\
&\leq \frac{\tilde{\eta}d}{2} + \frac{L^2\tilde{\eta}\beta d}{2}.
\end{aligned}
\tag{D.64}
$$

For the third term, if $B < \frac{|\sum_{m=1}^{M}(\boldsymbol{g}_m^{(t)})_i|}{\sigma}$, we have

$$
\begin{aligned}
&-\mathbb{E}\Bigg[< \nabla F(w^{(t)}), \frac{1}{M+B}\Bigg[\sum_{m=1}^{M} \textit{dp-sign}(\boldsymbol{g}_m^{(t)};\epsilon,\delta) + \sum_{j=1}^{B} \textit{byzantine-sign}(\boldsymbol{g}_j^{(t)})\Bigg] > \Bigg] \\
&= -\mathbb{E}\Bigg[\sum_{i=1}^{d} \nabla F(w^{(t)})_i \frac{1}{M+B}\Bigg[\sum_{m=1}^{M} \textit{dp-sign}((\boldsymbol{g}_m^{(t)})_i;\epsilon,\delta) + \sum_{j=1}^{B} \textit{byzantine-sign}((\boldsymbol{g}_j^{(t)})_i)\Bigg] > \Bigg] \\
&\leq -\sum_{i=1}^{d} |\nabla F(w^{(t)})_i| \frac{1}{M+B}\Bigg[\frac{|\sum_{m=1}^{M}(\boldsymbol{g}_m^{(t)})_i|}{\sigma} - B\Bigg] \\
&\leq -c||\nabla F(w^{(t)})||_1,
\end{aligned}
\tag{D.65}
$$

where c is some positive constant.

Following the same analysis as that in the proof of Theorem 20, the convergence of Algorithm 10 can be established. □

D.2 Discussions about *dp-sign* with $\delta = 0$

In this section, we present the differentially private compressor *dp-sign* with $\delta = 0$.

Definition 12. *For any given gradient $\boldsymbol{g}_m^t$, the compressor dp-sign outputs dp-sign$(\boldsymbol{g}_m^t, \epsilon, 0)$. In particular, the i-th entry of dp-sign$(\boldsymbol{g}_m^t, \epsilon, 0)$ is given by*

$$
\textit{dp-sign}(\boldsymbol{g}_m^t, \epsilon, 0)_i = \begin{cases} 1, & \textit{with probability } \frac{1}{2} + \frac{1}{2}\textit{sign}((\boldsymbol{g}_m^t)_i)\Big(1 - e^{-\frac{|(\boldsymbol{g}_m^t)_i|}{\lambda}}\Big), \\ -1, & \textit{with probability } \frac{1}{2} - \frac{1}{2}\textit{sign}((\boldsymbol{g}_m^t)_i)\Big(1 - e^{-\frac{|(\boldsymbol{g}_m^t)_i|}{\lambda}}\Big), \end{cases}
\tag{D.66}
$$

where $\lambda = \frac{\Delta_1}{\epsilon}$ and Δ_1 is the sensitivity measures defined in (5.11).

Theorem 24. *The proposed compressor dp-sign$(\cdot, \epsilon, 0)$ is $(\epsilon, 0)$-differentially private.*

Proof. Consider any vector $\boldsymbol{a}$ and $\boldsymbol{b}$ such that $||\boldsymbol{a} - \boldsymbol{b}||_1 \leq \Delta_1$ and $\boldsymbol{v} \in \{-1, 1\}^d$, we have

$$
\frac{P(\textit{dp-sign}(\boldsymbol{a}, \epsilon, 0) = \boldsymbol{v})}{P(\textit{dp-sign}(\boldsymbol{b}, \epsilon, 0) = \boldsymbol{v})} = \frac{\int_D e^{-\frac{||\boldsymbol{x}-\boldsymbol{a}||}{\lambda}} d\boldsymbol{x}}{\int_D e^{-\frac{||\boldsymbol{x}-\boldsymbol{b}||}{\lambda}} d\boldsymbol{x}},
\tag{D.67}
$$

where D is some integral area depending on $\boldsymbol{v}$. It can be verified that $e^{-\epsilon} \leq |\frac{e^{-\frac{||\boldsymbol{x}-\boldsymbol{a}||}{\lambda}}}{e^{-\frac{||\boldsymbol{x}-\boldsymbol{b}||}{\lambda}}}| \leq e^{\epsilon}$ always holds, which indicates that $e^{-\epsilon} \leq |\frac{P(\textit{dp-sign}(\boldsymbol{a},\epsilon,0)=\boldsymbol{v})}{P(\textit{dp-sign}(\boldsymbol{b},\epsilon,0)=\boldsymbol{v})}| \leq e^{\epsilon}$. □

Theorem 25. *Let $u_1, u_2, \cdots, u_M$ be M known and fixed real numbers. Further define random variables $\hat{u}_i = dp\text{-}sign(u_i, \epsilon, \delta), \forall 1 \leq i \leq M$. Then there always exist a constant σ_0 such that when $\sigma \geq \sigma_0$, $P(sign(\frac{1}{M}\sum_{m=1}^{M}\hat{u}_i) \neq sign(\frac{1}{M}\sum_{m=1}^{M}u_i)) < \left[\left(1-x\right)e^{x}\right]^{\frac{M}{2}}$, where $x = \frac{|\sum_{m=1}^{M}u_m|}{\gamma\lambda M}$ and γ is some positive constant.*

Proof. Without loss of generality, assume $u_1 \leq u_2 \leq \cdots \leq u_K < 0 \leq u_{K+1} \leq \cdots \leq u_M$ and $\frac{1}{M}\sum_{i=1}^{M}u_i < 0$. Note that similar analysis can be done when $\frac{1}{M}\sum_{i=1}^{M}u_i > 0$. Further define a series of random variables $\{X_i\}_{i=1}^{M}$ given by

$$X_i = \begin{cases} 1, & \text{if } \hat{u}_i \neq sign\Big(\frac{1}{M}\sum_{i=1}^{M}u_i\Big), \\ 0, & \text{if } \hat{u}_i = sign\Big(\frac{1}{M}\sum_{i=1}^{M}u_i\Big). \end{cases} \tag{D.68}$$

In particular, X_i can be considered as the outcome of one Bernoulli trial with successful probability $P(X_i = 1)$. Let $Z = \sum_{i=1}^{M}X_i$ and we have

$$P\left(sign\left(\frac{1}{M}\sum_{m=1}^{M}\hat{u}_i\right) \neq sign\left(\frac{1}{M}\sum_{m=1}^{M}u_i\right)\right) = P\left(Z \geq \frac{M}{2}\right). \tag{D.69}$$

Z follows the Poisson binomial distribution with mean and variance given by

$$\begin{aligned} \mu &= \sum_{m=1}^{M}P(X_m = 1) = \frac{M}{2} - \left[\sum_{m=1}^{K}P(u_m < n < 0) - \sum_{i=K+1}^{M}P(0 < n < u_m)\right], \\ \sigma^2 &= \sum_{m=1}^{M}P(n > -u_m)(1 - P(n > -u_m)), \end{aligned} \tag{D.70}$$

where $n \sim Laplace(0, \lambda)$. Similar to the analysis for *dp-sign* with $\delta > 0$, we can show that

$$\begin{aligned} &\sum_{m=1}^{K}P(u_m < n < 0) - \sum_{m=K+1}^{M}P(0 < n < u_m) \\ &= P\left(u_1 < n \leq -\sum_{m=2}^{M}u_m\right) + P\left(-\sum_{m=2}^{M}u_m < n < 0\right) - \sum_{m=2}^{M}P(0 < n < u_m) \\ &> \left|\sum_{m=1}^{M}u_m\right|\left[\frac{1}{2\lambda}e^{-\frac{|u_1|}{\lambda}}\right] + P\left(-\sum_{m=2}^{M}u_m < n < 0\right) - \sum_{m=2}^{M}P(0 < n < u_m) \\ &> \left|\sum_{m=1}^{M}u_m\right|\left[\frac{1}{2\lambda}e^{-\frac{|u_1|}{\lambda}}\right] - \left|\sum_{m=2}^{M}u_m\right|\left[\frac{1}{2\lambda}\left[1 - e^{-\frac{|\sum_{m=2}^{M}u_m|}{\lambda}}\right]\right] \\ &= \frac{1}{2\lambda}\left[\left|\sum_{m=1}^{M}u_m\right|e^{-\frac{|u_1|}{\lambda}} + \left|\sum_{m=2}^{M}u_m\right|\left[e^{-\frac{|\sum_{m=2}^{M}u_m|}{\lambda}} - 1\right]\right]. \end{aligned} \tag{D.71}$$

As a result, there exists a λ_0 such that when $\lambda \geq \lambda_0$, we have

$$\mu = \sum_{m=1}^{M} P(X_m = 1) \leq \frac{M}{2} + \frac{\sum_{m=1}^{M} u_m}{2\lambda\gamma}, \tag{D.72}$$

where γ is some constant larger than 1.

Following the same analysis as that in the proof of Theorem 13, we can show that there exists a positive constant M_0 such that when $M \geq M_0$

$$P\left(sign\left(\frac{1}{M}\sum_{m=1}^{M}\hat{u}_m\right) \neq sign\left(\frac{1}{M}\sum_{m=1}^{M}u_m\right)\right) = P\left(Z \geq \frac{M}{2}\right) < \left[\Big(1-x\Big)e^{x}\right]^{\frac{M}{2}}, \tag{D.73}$$

where $x = \frac{|\sum_{m=1}^{M} u_m|}{\gamma\lambda M}$ and γ is some positive constant. □

D.3 Discussions about the server's compressor $\frac{1}{M}sign(\cdot)$ in Algorithm 10

Recall that the update rule of the residual error is given by

$$\tilde{\boldsymbol{e}}^{t+1} = \frac{1}{M}\sum_{m=1}^{M} q(\boldsymbol{g}_m^t) + \tilde{\boldsymbol{e}}^t - \frac{a}{M}sign\left(\frac{1}{M}\sum_{m=1}^{M} q(\boldsymbol{g}_m^t) + \tilde{\boldsymbol{e}}^t\right), \tag{D.74}$$

where $a = 1$ in the proposed Algorithm.

Theorem 26. *In Algorithm 10, when the total number of workers is odd and $a = \frac{1}{2k+1}$ for any non-negative integer k, the server's compressor $\frac{a}{M}sign(\cdot)$ is an α-approximate compressor for some $\alpha > 0$.*

Proof. The goal is to show that $\frac{a}{M}sign(\cdot)$ is an α-approximate compressor, i.e., for $\boldsymbol{r}^t = \frac{1}{M}\sum_{m=1}^{M} q(\boldsymbol{g}_m^t) + \tilde{\boldsymbol{e}}^t$,

$$\left|\left|\boldsymbol{r}^t - \frac{a}{M}sign(\boldsymbol{r}^t)\right|\right|_2^2 = \sum_{i=1}^{d}\left(\boldsymbol{r}_i^t - \frac{a}{M}sign(\boldsymbol{r}_i^t)\right)^2 \leq (1-\alpha)||\boldsymbol{r}^t||_2^2 = (1-\alpha)\sum_{i=1}^{d}(\boldsymbol{r}_i^t)^2, \tag{D.75}$$

where $\boldsymbol{r}_i^t$ is the i-th entry of $\boldsymbol{r}^t$. In addition,

$$\left(\boldsymbol{r}_i^t - \frac{a}{M}sign(\boldsymbol{r}_i^t)\right)^2 = \left((\boldsymbol{r}_i^t)^2 - \frac{2a|\boldsymbol{r}_i^t|}{M} + \frac{a^2}{M^2}\right) = \left(1 - \frac{a}{M(\boldsymbol{r}_i^t)^2}\left(2|\boldsymbol{r}_i^t| - \frac{a}{M}\right)\right)(\boldsymbol{r}_i^t)^2. \tag{D.76}$$

It can be seen that a sufficient condition for (D.75) with some $\alpha > 0$ is given by

$$2|\boldsymbol{r}_i^t| - \frac{a}{M} > 0, \forall 1 \leq i \leq d. \tag{D.77}$$

Given that the total number of workers is odd, it is obvious that $(2k+1)M\boldsymbol{r}_i^0$ is an odd number. As a result, $(2k+1)M(\tilde{\boldsymbol{e}}^{(1)})_i = (2k+1)M\boldsymbol{r}_i^0 - sign(\boldsymbol{r}_i^0)$ is an even number. Similar to the proof of Lemma 9, it can be shown by induction that $(2k+1)M\boldsymbol{r}_i^t$ is odd for any t. Therefore,

$$2|\boldsymbol{r}_i^t| \geq \frac{2}{(2k+1)M} > \frac{a}{M}. \tag{D.78}$$

□

For $a \neq \frac{1}{2k+1}$, we consider four cases as follows.

Case 1: $a \geq 2$**.** In this case, since $(\tilde{\boldsymbol{e}}^0)_i = 0$, then if $\frac{1}{M}\sum_{m=1}^M q(\boldsymbol{g}_m^0)_i = \frac{1}{M}$, (D.77) is not satisfied.

Case 2: $1 < a < 2$**.** In this case, we consider a sequence of $\{\frac{1}{M}\sum_{m=1}^M q(\boldsymbol{g}_m^t)_i\}_{t=0}^n$, given by

$$\frac{1}{M}\sum_{m=1}^M q(\boldsymbol{g}_m^t)_i = \frac{1}{M}, \forall 0 \leq t \leq n. \tag{D.79}$$

Then

$$(\tilde{\boldsymbol{e}}^{(1)})_i = \frac{1-a}{M}, \quad \boldsymbol{r}_i^{(1)} = \frac{2-a}{M}. \tag{D.80}$$

Suppose that $2|\boldsymbol{r}_i^t| > \frac{a}{M}, \forall t$. Now we show that given $(\tilde{\boldsymbol{e}}^t)_i = \frac{t-ta}{M}, \boldsymbol{r}_i^t = \frac{t+1-ta}{M}$, and $a < \frac{t+1}{t}$, we have $(\tilde{\boldsymbol{e}}^{t+1})_i = \frac{t+1-t+1a}{M}, \boldsymbol{r}_i^{t+1} = \frac{t+2-t+1a}{M}$ and $a < \frac{t+2}{t+1}$. To satisfy $2|\boldsymbol{r}_i^t| > \frac{a}{M}$, we have $a < \frac{2t+2}{2t+1} < \frac{t+2}{t+1}$. In addition, according to (D.74), $(\tilde{\boldsymbol{e}}^{t+1})_i = \boldsymbol{r}_i^t - \frac{a}{M}sign(\boldsymbol{r}_i^t) = \frac{t+1-t+1a}{M}$. Then, $\boldsymbol{r}_i^{t+1} = \frac{t+2-t+1a}{M}$. As a result, by induction, we can show that $a < \frac{n+1}{n}$. As n increases, a decreases and approaches 1.

Case 3: $\frac{1}{2} < a < 1$**.** Again, we consider the sequence of $\{\frac{1}{M}\sum_{m=1}^M q(\boldsymbol{g}_m^t)_i\}_{t=0}^n$ given by (D.79). Then similarly, it can be shown that

$$(\tilde{\boldsymbol{e}}^t)_i = \frac{t-ta}{M}, \forall 1 \leq t \leq n+1. \tag{D.81}$$

Let $\frac{1}{M}\sum_{m=1}^M q(\boldsymbol{g}_m^{(n+1)})_i = -\frac{1}{M}$, we have

$$\boldsymbol{r}_i^{(n+1)} = \frac{1}{M}\sum_{m=1}^M q(\boldsymbol{g}_m^{(n+1)})_i + (\tilde{\boldsymbol{e}}^{(n+1)})_i = \frac{n-(n+1)a}{M}. \tag{D.82}$$

Taking $2|\boldsymbol{r}_i^{(n+1)}| > \frac{a}{M}$ yields

$$a \begin{cases} > \frac{2n}{2n+1} \geq \frac{n+1}{n+2}, & \text{if } a > \frac{n}{n+1}, \\ < \frac{2n}{2n+3}, & \text{if } a < \frac{n}{n+1}. \end{cases} \tag{D.83}$$

When $n = 1$, $\frac{n}{n+1} = \frac{1}{2}$. According to (D.83), $a > \frac{n}{n+1}$ is a sufficient condition of $a > \frac{n+1}{n+2}$. Therefore, as n increases, a increases and approaches 1.

Case 4: $\frac{1}{2k+2} < a \leq \frac{1}{2k}$, for any positive integer k. Again, we consider the sequence of $\{\frac{1}{M}\sum_{m=1}^{M} q(\boldsymbol{g}_m^t)_i\}_{t=0}^{n}$ given by (D.79). Similarly,

$$(\tilde{\boldsymbol{e}}^t)_i = \frac{t - ta}{M}, \forall 1 \leq t \leq n+1. \tag{D.84}$$

Let $\frac{1}{M}\sum_{m=1}^{M} q(\boldsymbol{g}_m^{(n+1)})_i = -\frac{s}{M}$, then we have

$$\boldsymbol{r}_i^{(n+1)} = \frac{1}{M}\sum_{m=1}^{M} q(\boldsymbol{g}_m^{(n+1)})_i + (\tilde{\boldsymbol{e}}^{(n+1)})_i = \frac{(n+1-s) - (n+1)a}{M}. \tag{D.85}$$

First of all, let $n = s = 2k - 1$, we have $\boldsymbol{r}_i^{(n+1)} = \frac{1-2ka}{M}$. Therefore, if $a = \frac{1}{2k}$, it is possible that $2|\boldsymbol{r}_i^{(n+1)}| = 0 \leq \frac{a}{M}$.

When $a < \frac{n+1-s}{n+1}$, $2|\boldsymbol{r}_i^{(n+1)}| > \frac{a}{M}$ yields

$$a < \frac{2(n+1-s)}{2n+3}. \tag{D.86}$$

Let $n = (2k+1) \times 2^y - 2$ and $s = 2k \times 2^y - 1$, where y is some non-negative integer, we have $\frac{n+1-s}{n+1} = \frac{2^y}{(2k+1)\times 2^y - 1}$. According to (D.86), $a < \frac{2^{y+1}}{(2k+1)\times 2^{y+1} - 1}$. When $y = 0$, $\frac{n+1-s}{n+1} = \frac{2^y}{(2k+1)\times 2^y - 1} = \frac{1}{2k}$. As y (and therefore n and s) increases, a decreases and approaches $\frac{1}{2k+1}$.

When $a > \frac{n+1-s}{n+1}$, $2|\boldsymbol{r}_i^{(n+1)}| > \frac{a}{M}$ yields

$$a > \frac{2(n+1-s)}{2n+1}. \tag{D.87}$$

Let $n = (2k+1) \times 2^y$ and $s = 2k \times 2^y + 1$, where y is some non-negative integer, we have $\frac{n+1-s}{n+1} = \frac{2^y}{(2k+1)\times 2^y + 1}$. According to (D.87), $a > \frac{2^{y+1}}{(2k+1)\times 2^{y+1} + 1}$. When $y = 0$, $\frac{n+1-s}{n+1} = \frac{2^y}{(2k+1)\times 2^y + 1} = \frac{1}{2k+2}$. As y (and therefore n and s) increases, a increases and approaches $\frac{1}{2k+1}$.

Remark 32. *By assuming that the total number of workers is odd, it is guaranteed that there is always a winner in the majority vote. When the number of workers is even, it is possible that $\boldsymbol{r}_i^t = 0$ and therefore (D.77) does not hold. This issue can be addressed if the server ignores the*

communication rounds (e.g., does not transmit anything) during which there is no winner in the majority vote.

D.4 Details of the Implementation

Our experiments are mainly implemented using Python 3.7.4 with packages tensorflow 2.0 and numpy 1.16.5. One Intel i7-9700 CPU with 32 GB of memory and one NVIDIA GeForce RTX 2070 SUPER GPU are used in the experiments.

D.4.1 Dataset and Pre-processing

We perform experiments on the standard MNIST dataset for handwritten digit recognition consisting of 60,000 training samples and 10,000 testing samples.[1] Each sample is a 28×28 size gray-level image. We normalize the data by dividing it with the max RGB value (i.e., 255.0).

D.4.2 Dataset Assignment

In our experiments, we consider 31 normal workers and measure the data heterogeneity by the number of labels of data that each worker stores. We first partition the training dataset according to the labels. For each worker, we randomly generate a set of size n which indicates the labels of training data that should be assigned to this worker. Then, a subset of training data from the corresponding labels is randomly sampled and assigned to the worker without replacement. The size of the subset depends on n and the size of the training data for each label. More specifically, we set the size of the subset as $\lfloor 60000/(31n) \rfloor$ in the beginning. When there are not enough training data for a label, we reduce the size of the subset accordingly. We consider the scenarios that all the workers have the same n. For the results in the third figure in Fig. 2, we set $n = 1, 2, 4$ for "1 LABEL", "2 LABELS", "4 LABELS", respectively. For the rest of the results, we set $n = 1$.

D.4.3 Neural Network Setting

We implement a two-layer fully connected neural network with softmax of classes with cross-entropy loss. The hidden layer has 128 hidden ReLU units.

D.4.4 Learning Rate Tuning

We use a constant learning rate η and tune the parameters from the set $\{1, 0.1, 0.01, 0.005, 0.001, 0.0001\}$. In particular, for Sto-SIGNSGD and EF-Sto-SIGNSGD, we set $\eta = 0.005$; for DP-SIGNSGD and

[1] Avaiable at `http://yann.lecun.com/exdb/mnist/`

EF-DP-SIGNSGD, we set $\eta = 0.01$. For FedAvg [1], we tune the learning rate from the set $\{2, 1.5, 1, 0.1, 0.01, 0.005, 0.001, 0.0001\}$, the number of local epochs from the set $\{1, 10, 20, 30\}$ and present the best result.

Appendix E

E.1 Proof of Theorem 22

Proof. According to Assumption 8, we have

$$
\begin{aligned}
&F(w^{(t+1)}) - F(w^{(t)}) \\
&\leq < \nabla F(w^{(t)}), w^{(t+1)} - w^{(t)} > + \frac{L}{2}||w^{(t+1)} - w^{(t)}||^2 \\
&= -\eta < \nabla F(w^{(t)}), sign(\sum_{m=1}^{M} \hat{g}_m^{(t)}) > + \frac{L}{2}||\eta sign(\sum_{m=1}^{M} \hat{g}_m^{(t)})||^2 \\
&= -\eta < \nabla F(w^{(t)}), sign(\sum_{m=1}^{M} \hat{g}_m^{(t)}) > + \frac{L\eta^2 d}{2} \\
&= \eta||\nabla F(w^{(t)})||_1 + \frac{L\eta^2 d}{2} - 2\eta \sum_{i=1}^{d} |\nabla F(w^{(t)})_i| \times \\
&\mathbb{1}_{sign(\sum_{m=1}^{M} \hat{g}_m^{(t)})_i = sign(\nabla F(w^{(t)})_i)},
\end{aligned}
\tag{E.1}
$$

in which $\nabla F(w^{(t)})_i$ is the i-th entry of the vector $\nabla F(w^{(t)})$. Taking expectation on both sides yields

$$
\begin{aligned}
&\mathbb{E}[F(w^{(t)}) - F(w^{(t+1)})] \geq -\eta||\nabla F(w^{(t)})||_1 - \frac{L\eta^2 d}{2} + 2\eta \times \\
&\sum_{i=1}^{d} |\nabla F(w^{(t)})_i| P\Big(\tilde{g}_i^{(t)} = sign(\nabla F(w^{(t)}))_i\Big).
\end{aligned}
\tag{E.2}
$$

□

E.2 Proof of Lemma 5

Proof. According to the constraint

$$1 - e^{-\frac{(2^{r_m}-1)N_0 B_m}{P_m}} \leq p_{out,m}, \tag{E.3}$$

when $\max\{r_m^{(1)}, r_m^{(3)}\} \leq r_m \leq r_m^{(2)}$, it can be obtained that

$$P_m \geq -\frac{N_0 B_m (2^{r_m}-1)}{\ln(1-p_{out,m})}. \tag{E.4}$$

Since the objective function $\frac{\alpha_m}{2} c_m D_m f_m^2 + \frac{P_m s_m}{r_m B_m}$ is an increasing function of P_m, we have

$$P_m^* = -\frac{N_0 B_m (2^{r_m}-1)}{\ln(1-p_{out,m})}. \tag{E.5}$$

According to the constraint

$$\frac{c_m D_m}{f_m} + \frac{s_m}{r_m B_m} \leq T_l, \tag{E.6}$$

we have

$$f_m \geq \frac{c_m D_m}{T_l - \frac{s_m}{r_m B_m}}. \tag{E.7}$$

In addition, the objective function is an increasing function of f_m. Therefore,

$$f_m^* = \max\left\{ \frac{c_m D_m}{T_l - \frac{s_m}{r_m B_m}}, f_{min,m} \right\} \tag{E.8}$$

□

E.3 Proof of Lemma 6

Proof. According to the constraint

$$\frac{\alpha_m}{2} c_m D_m f_m^2 + \frac{P_m s_m}{r_m B_m} \leq E_m, \tag{E.9}$$

we have

$$r_m \geq \frac{P_m s_m}{B_m (E_m - \frac{\alpha_m}{2} c_m D_m f_m^2)}. \tag{E.10}$$

According to the constraint

$$\frac{c_m D_m}{f_m} + \frac{s_m}{r_m B_m} \leq T_l, \tag{E.11}$$

we have

$$r_m \geq \frac{s_m f_m}{B_m f_m T_l - B_m c_m D_m}. \tag{E.12}$$

In addition, it can be shown that the objective function $\frac{M-2\sum_{m=1}^{M} p_{out}(r_m)}{\sqrt{T_l}}$ is a decreasing function of r_m. Therefore,

$$r_m^* = \max\left\{ \frac{P_m s_m}{B_m(E_m - \frac{\alpha_m}{2} c_m D_m f_m^2)}, \frac{s_m f_m}{B_m f_m T_l - B_m c_m D_m} \right\}. \tag{E.13}$$

□

E.4 Proof of Lemma 7

Proof. Define a series of random variables $\{\hat{X}_m\}_{m=1}^{M}$ given by

$$\hat{X}_m = \begin{cases} 1, & \text{if } sign(\hat{g}_m^{(t)})_i \neq sign(\nabla F_m(w^{(t)}))_i, \\ 0, & \text{if } sign(\hat{g}_m^{(t)})_i = sign(\nabla F_m(w^{(t)}))_i. \end{cases} \tag{E.14}$$

It can be verified that

$$P(\hat{X}_m = 1) = p_m^i p_{out}(r_m) + (1 - p_m^i)(1 - p_{out}(r_m)) = \frac{1}{2} - b|\nabla F_m(w^{(t)})_i|. \tag{E.15}$$

$$sign(\hat{g}_m^{(t)})_i = \begin{cases} 1, \text{ with probability } \frac{1+b\nabla F_m(w^{(t)})_i}{2}, \\ -1, \text{with probability } \frac{1-b\nabla F_m(w^{(t)})_i}{2}, \end{cases} \tag{E.16}$$

Further define a series of random variables $\{\hat{Z}_m\}_{m=1}^{M}$ given by

$$\hat{Z}_m = \begin{cases} 1, & \text{if } sign(\hat{g}_m^{(t)})_i = 1, \\ 0, & \text{if } sign(\hat{g}_m^{(t)})_i = -1. \end{cases} \tag{E.17}$$

Let $\hat{Z} = \sum_{m=1}^{M} \hat{Z}_m$, then

$$P\left(sign\left(\frac{1}{M} \sum_{m=1}^{M} sign(\hat{g}_m^{(t)})_i \right) = 1 \right) = P\left(\hat{Z} \geq \frac{M}{2} \right) = \sum_{H=\lceil \frac{M+1}{2} \rceil}^{M} P(\hat{Z} = H). \tag{E.18}$$

In addition, let $u_m = \nabla F_m(w^{(t)})_i$, we have

$$P(\hat{Z} = H) = \frac{\sum_{A \in F_H} \prod_{i \in A}(1 + bu_i) \prod_{j \in A^c}(1 - bu_j)}{2^M} = \frac{a_{M,H} b^M + a_{M-1,H} b^{M-1} + \cdots + a_{0,H} b^0}{2^M}, \tag{E.19}$$

in which F_H is the set of all subsets of H integers that can be selected from $\{1, 2, 3, ..., M\}$; $a_{m,H}, \forall 0 \leq m \leq M$ is some constant. It can be easily verified that $a_{0,H} = \binom{M}{H}$.

When b is sufficiently small, $P(\hat{Z} = H)$ is dominated by the last two terms in (E.19). In particular, $\forall m$, we have

$$\begin{aligned} \sum_{A \in F_H} \prod_{i \in A}(1 + bu_i) \prod_{j \in A^c}(1 - bu_j) = (1 + bu_m) &\sum_{A \in F_H, m \in A} \prod_{i \in A/\{m\}} (1 + bu_i) \prod_{j \in A^c} (1 - bu_j) \\ + (1 - bu_m) &\sum_{A \in F_H, m \notin A} \prod_{i \in A}(1 + bu_i) \prod_{j \in A^c/\{m\}} (1 - bu_j). \end{aligned} \tag{E.20}$$

As a result, when $\lceil \frac{M+1}{2} \rceil \leq H \leq M - 1$, the u_m related term in $a_{1,H}$ is given by

$$\left[\binom{M-1}{H-1} - \binom{M-1}{H} \right] u_m. \tag{E.21}$$

When $H = M$, the u_m related term in $a_{M-1,H}$ is given by

$$\left[\binom{M-1}{H-1} \right] u_m. \tag{E.22}$$

By summing over m, we have

$$a_{1,H} = \left[\binom{M-1}{H-1} - \binom{M-1}{H} \right] \sum_{m=1}^{M} u_m, \tag{E.23}$$

and

$$a_{1,H} = \left[\binom{M-1}{H-1} \right] \sum_{m=1}^{M} u_m, \tag{E.24}$$

when $\lceil \frac{M+1}{2} \rceil \leq H \leq M - 1$ and $H = M$, respectively.

By summing over H, we have

$$\sum_{H=\lceil \frac{M+1}{2} \rceil}^{M} a_{0,H} = \sum_{H=\lceil \frac{M+1}{2} \rceil}^{M} \binom{M}{H} = 2^{M-1}, \tag{E.25}$$

$$\sum_{H=\lceil\frac{M+1}{2}\rceil}^{M} a_{1,H} = \binom{M-1}{\lceil\frac{M+1}{2}\rceil - 1} \sum_{m=1}^{M} u_m \tag{E.26}$$

As a result,

$$\begin{aligned} P\left(\hat{Z} \geq \frac{M}{2}\right) = \sum_{H=\lceil\frac{M+1}{2}\rceil}^{M} P(\hat{Z}=H) &= \frac{2^{M-1} + \binom{M-1}{\lceil\frac{M+1}{2}\rceil-1} \sum_{m=1}^{M} u_m b}{2^M} + O\left(\frac{b^2}{2^M}\right) \\ &= \frac{1}{2} + \frac{\binom{M-1}{\lceil\frac{M+1}{2}\rceil-1}}{2^M} \sum_{m=1}^{M} u_m b + O\left(\frac{b^2}{2^M}\right), \end{aligned} \tag{E.27}$$

which completes the proof. □

E.5 Proof of Theorem 23

Proof. It is shown in Theorem 22,

$$\begin{aligned} \mathbb{E}[F(w^{(t)}) - F(w^{(t+1)})] \geq &- \eta ||\nabla F(w^{(t)})||_1 - \frac{L\eta^2 d}{2} \\ &+ 2\eta \sum_{i=1}^{d} |\nabla F(w^{(t)})_i| P\Big(\tilde{g}_i^{(t)} = sign(\nabla F(w^{(t)}))_i\Big). \end{aligned} \tag{E.28}$$

According to Lemma 7 and Remark 31, $P\Big(\tilde{g}_i^{(t)} = sign(\nabla F(w^{(t)}))_i\Big) > 1/2$. Therefore, there always exists a positive constant c such that

$$P\Big(\tilde{g}_i^{(t)} = sign(\nabla F(w^{(t)}))_i\Big) \geq 1/2 + c. \tag{E.29}$$

Plugging (E.29) into (E.28) yields

$$\begin{aligned} \mathbb{E}[F(w^{(t)}) - F(w^{(t+1)})] &\geq -\eta ||\nabla F(w^{(t)})||_1 - \frac{L\eta^2 d}{2} + 2\eta \sum_{i=1}^{d} |\nabla F(w^{(t)})_i| \left(\frac{1}{2} + c\right) \\ &= 2\eta c ||\nabla F(w^{(t)})||_1 - \frac{L\eta^2 d}{2}. \end{aligned} \tag{E.30}$$

Adjusting the above inequality and averaging both sides over $t = 1, 2, \cdots, T$, we can obtain

$$\frac{1}{T} \sum_{t=1}^{T} \eta c ||\nabla F(w^{(t)})||_1 \leq \frac{\mathbb{E}[F(w^{(0)}) - F(w^{(T+1)})]}{T} + \frac{L\eta^2 d}{2}. \tag{E.31}$$

Letting $\eta = \frac{1}{\sqrt{dT}}$ and dividing both sides by η gives

$$\frac{1}{T}\sum_{t=1}^{T} c||\nabla F(w^{(t)})||_1 \leq \frac{\mathbb{E}[F(w^{(0)}) - F(w^{(T+1)})]\sqrt{d}}{\sqrt{T}}, \tag{E.32}$$

which completes the proof. □