ABSTRACT

HE, XIAOFAN. Surviving the Information Warfare: from Static Competition to Dynamic Game. (Under the direction of Dr. Huaiyu Dai.)

In view of the pervasive applications of wireless technologies, wireless security is becoming increasingly important. In this dissertation, we explore some important and timely problems in wireless security.

The first part of this dissertation is devoted to addressing the Byzantine attack in cognitive radio (CR) networks. In the Byzantine attack, malicious secondary users can send falsified local spectrum inference to mislead the fusion center’s decision, so as to prevent honest secondary users from using the existing white space, or allure them to access the channels in use and cause excessive interference to legitimate users, thereby undermining the premise of CR technology. To resist the Byzantine attack, two novel defense schemes are proposed in Chapter 2 and Chapter 3, respectively. In Chapter 2, the hidden Markov model (HMM) is employed to characterize users’ sensing behaviors, and a new HMM estimation algorithm that can jointly estimate multiple correlated HMMs using only unlabeled data is developed. Subsequently, this algorithm is applied to separate malicious users from honest ones, providing an effective defense against the Byzantine attack. In Chapter 3, another countermeasure to the Byzantine attack, termed Conditional Frequency Check (CFC), is proposed, in which two natural but effective CFC statistics that explore the second-order properties of the underlying Markov spectrum model are constructed. Through both analytical and numerical analysis, it is shown that the proposed CFC statistics clearly manifest the statistical difference of the sensing behaviors of the malicious and honest users, offering another effective defense against the Byzantine attack. As compared to existing works in literature, both of our proposed methods can handle more general malicious behaviors and do not require any prior knowledge of the true spectrum. Also, the HMM estimation algorithm proposed in Chapter 2 is rather general and may find wider applications in other relevant fields, while the CFC developed in Chapter 3 is tailor-designed for Byzantine attack detection and thus achieves even better performance.

Besides the static Byzantine attack discussed above, dynamic and adaptive attacks in which the adversary can astutely change their attacking strategies according to the defense mechanisms also appear, escalating conventional security problems to more sophisticated dynamic security competitions. To survive such escalated information warfare, more effective defenses and advanced analytical tools are needed. In Chapter 4, a dynamic mobile jamming/anti-jamming competition, where both legitimate relay nodes and jammers can masterly move to more advantageous positions based on the opponents’ current locations, is considered to exemplify the envisioned dynamic security competitions. Existing approaches for directing node
movement in such scenarios are mainly based on heuristics and simulation studies, and thus lack performance assurance. Considering this, new spectral quantities and corresponding eigenvalue variants based on spectral graph theory are proposed in Chapter 4 for node mobility control. As compared to existing literature, the proposed method is built on a more solid mathematical foundation and thus offers better performance.

Following further along this line, Chapters 5–7 are dedicated to more involved dynamic stochastic security games, where the defender and the attacker can not only adapt to the opponent’s current strategy but also think ahead and conduct foresighted planning. Game theory serves as a natural tool to study such complex interactions. However, classical game theory usually assumes a static environment, while the arms race between the defender and attacker in practice often takes place in unknown dynamic environments. In such cases, the stochastic game (SG) is a more suitable framework for the corresponding analysis and strategy design. Multiagent reinforcement learning (MARL) algorithms that enable a player to gradually learn good strategies through trial-and-error interactions with both the opponent and the unknown dynamic environment are often employed for SG in practice. Nonetheless, conventional MARL assumes equally knowledgeable players, while information asymmetry often exists in practical security games. Particularly, two types of information asymmetry are considered in our works. The first type of information asymmetry is presented when the defender holds extra local information that is unknown to the attacker, while the second type corresponds to the case that the defender has uncertainties about the attacker. In Chapter 5, two new MARL algorithms, termed minimax-PDS and WoLF-PDS, are proposed that enable the defender to expedite its learning and adaptation when the first type of information asymmetry exists. To deal with the second type of information asymmetry in security games, a joint reinforcement learning (RL) and type identification algorithm and a Bayesian Nash-Q learning algorithm are proposed in Chapter 6 and Chapter 7 for the cases of zero-sum and non-zero-sum security games, respectively. By using these algorithms, the defender can gradually infer the unknown information about the attacker based on the observed attacker actions and then adjust its defense strategy accordingly.

Besides developing new algorithms, correct understanding and comprehensive evaluation of newly developed security techniques are also critical. Considering this, recently advocated link signature based security mechanisms are reexamined in Chapter 8. Particularly, these are practical security protocols that aim at establishing secure communication between wireless devices by exploiting wireless channel characteristics. In current literature of link signature, it is commonly believed that the corresponding wireless channels are essentially uncorrelated as long as adversary sensors are kept away from the legitimate system by more than half a wavelength; however, some contradictory facts have been observed in other related works. To avoid false sense of security, the dependability of the link signature itself deserves further in-
vestigation. To this end, existing wireless channel models are revisited in Chapter 8, with a focus on exploring the corresponding security implications to link signature. Several key factors that have significant influence on link signature security have been identified, and based on the obtained understanding, a generic channel correlation model is developed for link signature, which facilitates the study on the guard zone deployment in typical indoor and outdoor communication scenarios. Real-world experiments based on Universal Software Radio Peripheral (USRP) platforms and GNURadio are also conducted to further justify the analysis.

In summary, several interesting and important security problems in current and emerging wireless systems have been investigated in this dissertation. It is our hope that our work may stimulate further development in this burgeoning research field.
Surviving the Information Warfare: from Static Competition to Dynamic Game

by

Xiaofan He

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

2015

APPROVED BY:

Dr. Dror Baron

Dr. Brian Hughes

Dr. Rudra Dutta

Dr. Keith Townsend

Dr. Huaiyu Dai
Chair of Advisory Committee
DEDICATION

To my family.
BIOGRAPHY

Xiaofan He received his Bachelor of Engineering degree in Electronics and Information Engineering in 2008 from Huazhong University of Science and Technology, Wuhan, China; and Master of Science degree in Electrical and Computer Engineering in 2010 from McMaster University, Hamilton, ON, Canada. In August 2011, he started his work towards the PhD degree in Electrical Engineering at North Carolina State University, Raleigh, NC, under the guidance of Prof. Huaiyu Dai. His doctoral research mainly focuses on wireless and network security.
ACKNOWLEDGEMENTS

First, I would like to express my sincere gratitude to my advisor, Professor Huaiyu Dai, for his patient guidance and continuous support to my doctoral research adventure and future career development. Besides, he also sets an excellent example of integrity, gentleness and humility for me in life. It is my fortune to work with him and learn from him.

My gratitude also goes to Professor Dror Baron, Professor Rudra Dutta, Professor Brian Hughes, Professor Peng Ning, Professor Munindar Singh and Professor Keith Townsend, for their illuminating instruction, encouragement, and inspiring advice. I would like to take the opportunity to thank all professors and teachers whose courses have provided the solid background and foundation for my thesis research as well.

In addition, I am grateful to my colleagues and friends, Chengzhi, Juan, Huazi, Junghoon, Nakul, Yufan, Srinjoy, Dong, Yang, Hao, Richeng, Ali, Wenbo, Yao, Fengyuan, Yuan, Feier, Jin, Yanting, Junan and Wenxing. Their enlightening discussions and invaluable friendship have been continuous fuel during my long journey in finishing this doctoral program.

Furthermore, I would like to thank my dear parents. Their unconditional love and support fill me with courage and strength to pursue my dream, and every progress I made in my life owes to them. Finally, a special thank goes to my beloved Juan. Life would not have been as colorful without her love, her companionship, and all the hope and joy she brought.
# TABLE OF CONTENTS

LIST OF TABLES ......................................................... ix

LIST OF FIGURES ....................................................... x

Chapter 1 Introduction ................................................ 1
  1.1 Combating Byzantine Attackers in CR Networks ................. 1
  1.2 Dynamic Security Competition .................................. 3
  1.3 Evaluation of Link Signature .................................. 5
  1.4 Organizations ................................................... 7

Chapter 2 HMM-Based Malicious User Detection for Robust Collaborative Spectrum Sensing ........................................... 8
  2.1 Problem Formulation ............................................. 9
    2.1.1 System Model and Spectrum Sensing Metrics .............. 9
    2.1.2 Two Correlated HMMs ...................................... 11
  2.2 The Proposed Method ........................................... 12
    2.2.1 Derivations of the Proposed Estimation Method .......... 12
    2.2.2 Convergence Analysis .................................... 17
    2.2.3 Implementation .......................................... 18
    2.2.4 Secondary User Classification ............................ 20
  2.3 Hard- and Soft Data Fusion .................................... 22
  2.4 Simulations ..................................................... 23
    2.4.1 A Basic Example ......................................... 23
    2.4.2 Further Simulations ...................................... 25
    2.4.3 Comparison of Hard- and Soft Data Fusion .............. 27
  2.5 Conclusions ..................................................... 29

Chapter 3 A Byzantine Attack Defender in Cognitive Radio Networks: the Conditional Frequency Check .................................. 30
  3.1 Problem Formulation ............................................. 31
  3.2 The Proposed Method ........................................... 32
    3.2.1 Conditional Frequency Check .............................. 32
    3.2.2 The Hamming Distance Check .............................. 36
  3.3 Selection of Thresholds ......................................... 37
    3.3.1 Selection of $\beta_{CFC}$ ................................ 38
    3.3.2 Selection of $\beta_{HDC}$ ................................ 38
  3.4 Extensions of the CFC ........................................... 38
    3.4.1 More General Attackers .................................... 39
    3.4.2 In the Absence of Trusted Sensor ........................ 41
  3.5 Simulations ..................................................... 43
    3.5.1 Basic Examples ............................................ 43
    3.5.2 Further Simulations ...................................... 46
    3.5.3 Malicious Users with More/Less Powerful Sensors ....... 49
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Section</th>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5.4</td>
<td>Removing the Trusted Sensor</td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td>3.6</td>
<td>Conclusions</td>
<td></td>
<td></td>
<td>50</td>
</tr>
<tr>
<td>4.1</td>
<td>Problem Formulation and Notations</td>
<td></td>
<td></td>
<td>52</td>
</tr>
<tr>
<td>4.2</td>
<td>Single Flow under Jamming</td>
<td>4.2.1</td>
<td>Preliminary on the Cheeger Constant</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.2.2</td>
<td>A Naive Approach</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.2.3</td>
<td>Single-weighted Cheeger Constant</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.2.4</td>
<td>Mobility Control Based on (\lambda_2(LW))</td>
<td>57</td>
</tr>
<tr>
<td>4.3</td>
<td>Multiple Flows under Jamming</td>
<td>4.3.1</td>
<td>The Multicommodity Flow Problem</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.3.2</td>
<td>Multi-weighted Cheeger Constant</td>
<td>58</td>
</tr>
<tr>
<td>4.4</td>
<td>Numerical Results and Discussion</td>
<td>4.4.1</td>
<td>Single Commodity Flow</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.4.2</td>
<td>Multicommodity Flow</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.4.3</td>
<td>Application in Cognitive Radio Networks</td>
<td>70</td>
</tr>
<tr>
<td>4.5</td>
<td>Related Works</td>
<td></td>
<td></td>
<td>72</td>
</tr>
<tr>
<td>4.6</td>
<td>Conclusions</td>
<td></td>
<td></td>
<td>72</td>
</tr>
<tr>
<td>5.1</td>
<td>Problem Formulation and Background</td>
<td></td>
<td></td>
<td>78</td>
</tr>
<tr>
<td>5.1.1</td>
<td>Problem Formulation</td>
<td></td>
<td></td>
<td>78</td>
</tr>
<tr>
<td>5.1.2</td>
<td>Conventional MARL</td>
<td></td>
<td></td>
<td>79</td>
</tr>
<tr>
<td>5.1.3</td>
<td>Limitation of Conventional MARL</td>
<td></td>
<td></td>
<td>81</td>
</tr>
<tr>
<td>5.2</td>
<td>Multi-agent PDS Learning</td>
<td>5.2.1</td>
<td>Minimax-PDS</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.2.2</td>
<td>WoLF-PDS</td>
<td>83</td>
</tr>
<tr>
<td>5.3</td>
<td>Applications in Practical Anti-jamming Games</td>
<td>5.3.1</td>
<td>Anti-jamming in EHCS</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.3.2</td>
<td>Anti-jamming in CR Systems</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.3.3</td>
<td>Cloud-based Security Game</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.3.4</td>
<td>Applications of the Proposed Algorithms</td>
<td>89</td>
</tr>
<tr>
<td>5.4</td>
<td>Simulations</td>
<td>5.4.1</td>
<td>EHCS Anti-jamming</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.4.2</td>
<td>SU Anti-jamming</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.4.3</td>
<td>Cloud-based Security Game</td>
<td>95</td>
</tr>
<tr>
<td>5.5</td>
<td>Related Works</td>
<td></td>
<td></td>
<td>96</td>
</tr>
<tr>
<td>5.6</td>
<td>Conclusion</td>
<td></td>
<td></td>
<td>97</td>
</tr>
<tr>
<td>6.1</td>
<td>Problem Formulation and Background</td>
<td></td>
<td></td>
<td>99</td>
</tr>
<tr>
<td>6.1.1</td>
<td>General Stochastic Game and the Spectrum Access Game</td>
<td></td>
<td></td>
<td>99</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 2.1 Notations and symbols ......................................................... 10
Table 2.2 Comparison of $\epsilon(B)$ using different $l$’s in (2.17) and (2.18). ........ 23
Table 2.3 Comparison of HMM parameter estimation errors. ................................. 24
Table 2.4 Performance comparison for the three fusion centers. .............................. 24

Table 3.1 Average performance comparison over 1000 runs. ............................... 44
Table 3.2 Average performance comparison over 1000 runs for more general malicious sensors. ................................................................. 49
Table 3.3 Performance of the proposed method without the trusted sensor over 1000 runs. 50
Table 3.4 Performance of the proposed method without the trusted sensor over 1000 runs when malicious sensors are more powerful. 50

Table 4.1 Comparison of flows and power efficiencies. ....................................... 63

Table 8.1 Important factors for LS security. (O: one-ring, T: two-ring, E: elliptical scatterer-ring, F: far scatterer-ring) ................................. 135
Table 8.2 Scenario I .................................................................................. 138
Table 8.3 Scenario II .................................................................................. 139
Table 8.4 Scenario III .................................................................................. 140
Table 8.5 Indoor Experiment ......................................................................... 141
Table 8.6 Outdoor Experiment ........................................................................ 144
LIST OF FIGURES

Figure 2.1 Single user spectrum sensing model. ................................................. 10
Figure 2.2 Collaborative spectrum sensing over $T$ time slots. .............................. 11
Figure 2.3 The block diagram of the proposed algorithm. ................................. 22
Figure 2.4 Comparison of spectrum sensing performances over 100 Monte Carlo runs. . 24
Figure 2.5 Spectrum sensing and classification performances comparisons ($L = 20$, $P^H_d = 0.85$ and $P^H_{fa} = 0.15$). The shaded region corresponds to the method in [1]. .............................................. 26
Figure 2.6 Malicious user detection performance for different $L$ ($\text{SSA}_{th} = 0.1$, $P^H_d = 0.85$ and $P^H_{fa} = 0.15$). .............................................. 28
Figure 2.7 Hard- and soft data fusion comparison with perfect $\lambda$ and $m$ ($L = 20$). . 28
Figure 2.8 Hard- and soft data fusion comparison with estimated $\hat{\lambda}$ and $\hat{m}$ ($L = 20$). . 28
Figure 3.1 Typical graph of $e(\phi)$ when the condition $\frac{a_10 P_{fa} + a_01 P_d}{a_10 + a_01} = \frac{1}{2}$ holds and $DM > 0$. 36
Figure 3.2 Typical graph of $e(\phi)$ when the condition $\frac{a_10 P_{fa} + a_01 P_d}{a_10 + a_01} = \frac{1}{2}$ holds and $DM = 0$. 37
Figure 3.3 The block diagram of the proposed algorithm. ................................. 43
Figure 3.4 Malicious sensor detection result using CFC in the first case. .................. 44
Figure 3.5 Malicious sensor detection result using CFC and HDC in the first case. .... 45
Figure 3.6 True sensor types in the first case. .................................................. 45
Figure 3.7 Malicious sensor detection result using CFC in the second case. ............... 45
Figure 3.8 True sensor types in the second case. ................................................ 46
Figure 3.9 Spectrum sensing performance at the fusion center in scenario 1. .......... 46
Figure 3.10 Spectrum sensing performance at the fusion center in scenario 2. ......... 47
Figure 3.11 Spectrum sensing performance comparison in scenario 3. .................... 48
Figure 3.12 Comparison of classification accuracy of the proposed method in all scenarios. 48

Figure 4.1 $u(\cdot)$ ($\xi = 1$ and $\kappa = 10$). .................................................. 52
Figure 4.2 Single flow and static jammer. ....................................................... 61
Figure 4.3 Single flow and mobile jammer. ..................................................... 62
Figure 4.4 Single flow using the method in [2]. .............................................. 64
Figure 4.5 Comparison of average $F_{sgl}$’s under different jammer localization errors. . 64
Figure 4.6 Comparison of $F_{sgl}$’s under different jammer moving frequencies. ....... 65
Figure 4.7 Performance comparison for Rayleigh fading channels. ...................... 66
Figure 4.8 Using $\lambda_2(LW)$ and mobile destination. ...................................... 67
Figure 4.9 Initial positions. ............................................................................. 68
Figure 4.10 Comparison of $F_{mult}$’s for 2-unicast (static jammer). ...................... 68
Figure 4.11 Comparison of relay final positions for 2-unicast (static jammer). ........ 69
Figure 4.12 Trajectories for 2-unicast (mobile jammers). ................................... 70
Figure 4.13 Comparison of $F_{mult}$’s for 2-unicast (mobile jammer). .................. 71
Figure 4.14 Comparison of relay final positions for 1-to-3 multicast (static jammer). .... 71
Figure 4.15 Comparison of $F_{mult}$’s for 1-to-3 multicast (static jammer). .......... 72
Figure 4.16 Trajectories for 1-to-3 multicast (mobile jammers). ......................... 73
Figure 4.17 Comparison of $F_{mult}$’s for 1-to-3 multicast (mobile jammers). ........ 74
Chapter 1

Introduction

Wireless communication technologies are permeating all aspects of our society, from family life and social interactions to national infrastructures and military applications, and in many scenarios, such as emergency rescue and power networks, the availability and quality of wireless communication is among the top concerns. However, while bringing unprecedented convenience and productivity to our lives, wireless systems have also exhibited vast vulnerabilities for the adversaries to explore, leaving its security an ever present concern. In this dissertation, we focus on developing countermeasures to the Byzantine attack in cognitive radio (CR) networks, studying effective mobility control against adaptive mobile jammers, designing new learning algorithms in dynamic security games with information asymmetry, and evaluating the dependability of wireless link signature. All these are crucial and timely security issues for current and emerging wireless systems.

1.1 Combating Byzantine Attackers in CR Networks

With the ever increasing demand for the scarce wireless spectrum, cognitive radio has been proposed as a viable and promising way of improving spectrum efficiency. Spectrum sensing is one of the essential functions that determine the success of the cognitive radio technology. Various collaborative spectrum sensing schemes have been proposed to overcome the fallibility of single user spectrum sensing due to channel uncertainties, and consequently improve the sensing performance [3, 4]. However, collaborative spectrum sensing also induces security vulnerabilities. In particular, malicious sensors can send falsified local spectrum inference to mislead the fusion center’s decision, so as to prevent honest secondary users from using the existing white space, or allure them to access the channels in use and cause excessive interference to legitimate users, thereby undermining the premise of CR technology. This is known as the Byzantine attack [1] (a.k.a. spectrum sensing data falsification (SSDF) attack [5]).
To resist the Byzantine attack, several countermeasures have been proposed in literature, where the essential idea is to detect the attacker according to its abnormal behavior. In [6], a weighted sequential probability ratio test is proposed for robust distributed spectrum sensing. In [7], out-lier factors are computed for secondary users to identify malicious ones. In [8], suspicious levels of all nodes are computed and the nodes with a high suspicious level are excluded from decision-making to ensure robustness. All these methods are based on the concept of reputation. One key feature of this type of methods is that lower reputations will be assigned to sensors which provide inconsistent spectrum inference as opposed to the majority’s decisions. However, the success of this type of methods is built upon the assumption that the global decision is correct, which may not be true when malicious sensors dominate the network. In fact, it was shown in [1, 9] that when the number of Byzantine attackers in the network exceeds a certain fraction, such reputation based methods become completely ineffective.\(^1\) Non-reputation based approaches have also been developed. For example, in [10], a double-sided neighbor distance metric and the frequency check (FC) method were proposed to detect malicious users. In [11], a correlation filter approach, which relies on the fact that the received signal strengths among nearby secondary users are correlated, is proposed to mitigate the attack from malicious users. However, these methods still rely on the correctness of the global decision and thus are only applicable to scenarios where a small fraction of users are malicious. When the majority of users may not be trustworthy, global decision independent approaches are more suitable. Works in this line include the prior-probability aided method proposed in [12, 13] and the user-centric misbehavior detection method presented in [14]. All these methods either explicitly consider the i.i.d. spectrum state model for simplicity [10, 12, 14], or focus their analysis on one time slot and ignore the correlation between the spectrum states [1, 6–8, 15, 16]. But in many scenarios, the primary channel states are correlated in time, and it is more accurate to model them as Markovian [17–21]. In [22, 23], it has already been shown that better spectrum sensing performance can be obtained when the Markovian model is adopted, though all secondary users are assumed to be trustworthy and the security aspect of the spectrum sensing problem is ignored there. On the other hand, there is a scarcity in literature of malicious user detection methods under the Markov model. For such a setting, a malicious user detection algorithm based on the non-parametric Kruskal-Wallis test is developed in [16], but the considered attack is restricted to one-way flipping (i.e., changing the sensing result from idle to occupied or the other way around). Also, in [13], a Markovian spectrum model is considered but the proposed malicious user detection algorithm requires prior knowledge of the state transition probabilities, which is difficult to obtain in the presence of attackers.

In Chapter 2, a novel HMM estimation algorithm that can jointly estimate multiple correlated HMMs using only unlabeled data is developed. By modeling honest users and malicious

\(^1\)When all sensors have the same spectrum sensing ability, the fraction is 50% [1].
users as two HMMs, the proposed algorithm can separate all the users into two groups based on their different statistical behaviors. Then two auxiliary tests are further invoked to identify the malicious group based on the corresponding HMM estimates. As compared to the previous works with Markovian spectrum models, the proposed method can defend against malicious users that flip local sensing results in both directions, and eliminates the requirement of prior knowledge about the true spectrum. It is also worth mentioning that, the proposed HMM inference method is rather general and may find applications in other relevant fields.

In Chapter 3, another countermeasure to the Byzantine attack, termed Conditional Frequency Check (CFC), which can further improve the detection accuracy is proposed. In particular, two natural but effective CFC statistics, which explore the second order properties of the assumed Markovian spectrum model, are constructed. Through analysis, we show that the CFC statistics can serve as effective metrics for separating malicious users from honest ones. With the aid of one trusted sensor and a sufficiently long detection window, the proposed CFC method is capable of detecting any malicious sensor regardless of the proportion of malicious ones in the sensor group, without requiring prior knowledge of the true spectrum. In the case when such a trusted sensor is not available, an auxiliary clustering procedure is developed, which can effectively detect the malicious sensors as long as honest sensors constitute the majority.

1.2 Dynamic Security Competition

The Byzantine attack discussed in Chapter 2 and Chapter 3 is a static attack, in the sense that the adversary is assumed not able to change its strategy according to the defense mechanism. Recently, more intelligent and adaptive attacks also appear due to the advancement in both software and hardware, escalating conventional static security problems to more sophisticated dynamic security competitions. To survive such escalated information warfare, more effective defenses and advanced analytical tools are needed.

The rivalry between a legitimate mobile network and a group of mobile jammers considered in [2] serves as a good example for dynamic security competition, where both the legitimate nodes and the jammers can reconfigure their locations in response to the opponent’s action; however, the study in [2] is mainly based on inspiring heuristics and simulations. To provide performance assurance in such dynamic jamming/anti-jamming competitions, a mobility control scheme based on spectral graph theory [24] is developed in Chapter 4. Particularly, the scenarios that a set of legitimate mobile relays fight against a set of mobile jammers so as to increase the achievable network flows are considered. The dynamic adaptive competition between relays and jammers evolves as follows: relay nodes move based on jammers’ current positions to protect the network flows, while the jammers strive in opposition and adaptively move to new advantageous positions so as to degrade the achievable network flows. For such dynamic competitions, new
spectral quantities, single- and multi-weighted Cheeger constants and corresponding eigenvalue variants, are constructed so that relays and jammers can compete effectively by directing their motions according to these spectral quantities of the network graph. In addition, the proposed scheme also finds an interesting application in mobile CR networks. Particularly, by treating primary users (PU) as unintended jamming sources, the proposed scheme can be employed to guide the movements of mobile secondary users according to PU location changes. Moreover, it is worth mentioning that the proposed scheme conforms to the newly advocated moving target defense (MTD) principle [25], which indicates that the ability of dynamic adaptation is crucial to the success in the security contest. This principle has already been exemplified in the cyber domain and the spectrum domain. Cyber MTD approaches include software transformation, server diversification, IP evolution and key maneuvering [25]. In the spectrum domain, legitimate transceivers can also switch the attack surface through various channel hopping techniques [26–28]. In this sense, anti-jamming via controlled mobility provides a new avenue to instantiate the MTD principle.

Following further along this line, it is also interesting to study dynamic security games where the defender and the attacker can not only adapt to the opponent’s current strategy but also think ahead and conduct foresighted planning. For such problems, game theory, which provides a formal analytical framework to study the complex interactions among rational players, seems a natural tool to consider. Recently, there has been significant growth in research activities that use game theory to analyze network security problems (e.g., [29, 30] and references therein), and both the simultaneous-move game model (e.g., [31]) and the sequential-move (Stackelberg) game model (e.g., [32]) have been considered. Most of these works employ classical game theory which usually assumes a static environment. However, the arms race between the defender and attacker in practice often takes place in (unknown) dynamic environments. Changes in wireless environments and variations of available communication and computational resources are possible sources of such dynamics. This suggests that, in practical applications, legitimate systems have to not only meticulously deal with the intelligent adversary but also carefully accommodate their strategies in accordance to these environmental dynamics. With this consideration, our study on the dynamic security games in Chapters 5–7 adopt the stochastic game (SG) framework [33] which can explicitly capture the environment dynamics in the corresponding game models. For environment with unknown dynamics in practice, both the legitimate system and the adversary can employ multi-agent reinforcement learning (MARL) algorithms [33] to gradually learn their (ideally optimal) strategies in the corresponding SG.

Although SG together with MARL can address the environmental dynamics in security games in a systematic manner, there is a notable mismatch between their underlying modeling and practice: Conventional MARL algorithms usually assume that both the legitimate system and the adversary are equally knowledgeable about the ongoing competitions, while in
practical security games, information asymmetry between the defender and the attacker usually exists. Particularly, two types of information asymmetry are considered in this dissertation. The first type of information asymmetry presents when the defender holds extra local information that is unknown to the attacker. For example, an energy harvesting communication system (EHCS) preserves information about its energy harvesting process; a SU holds its own data arrival statistics and transmission schedules; and a cloud client keeps its own statistics about the cloud resource availability. In practice such information is unlikely to be known accurately by the adversary. With this observation, two new MARL algorithms, termed minimax-PDS and WoLF-PDS, are proposed in Chapter 5 to enable the defense system to learn and adapt faster in unknown dynamic environments by exploiting such information advantage. Through analysis, it is shown that these two algorithms are provably convergent and rational, respectively. The second type of information asymmetry corresponds to the case that the defender has uncertainties about (e.g., the existence or the objective of) the attacker. For instance, in a wireless communication network, malicious users may disguise themselves using faked ID or through compromising legitimate users; an intrusion detection system (IDS) may encounter different intruders whose exact purposes are seldom known accurately beforehand - sometimes the attacks may be merely due to misbehaviors of a negligent legitimate user. When the environment is static, Bayesian game is the common approach used in literature (e.g., [34–37]) to deal with this type of information asymmetry. To handle this type of information asymmetry in more complicated dynamic environments, a joint reinforcement learning and type identification algorithm is proposed in Chapter 6 for zero-sum dynamic security games. To further address the non-zero-sum cases, a Bayesian Nash-Q learning algorithm is developed in Chapter 7. The underlying rationale of these two algorithms is that the actions of the opponent are type-driven and hence can be exploited to facilitate the inference of the unknown information about the opponent.

1.3 Evaluation of Link Signature

Besides developing new algorithms, another important aspect in the security research is to obtain comprehensive evaluation of newly developed security techniques so as to avoid false sense of security. Considering this, the dependability of the recently advocated link-signature based security mechanisms is investigated in Chapter 8. Particularly, link signature refers to the (unique) channel characteristics between two wireless devices, based on which secure communications can be established. This idea is originated from the concept of information-theoretic security [38–43]. Among various applications of link signature, secret key extraction and location distinction (a.k.a. physical layer authentication) are two prominent ones. In [44], a theoretical scheme for generating secret bits from correlated observations of deep channel fades is pro-
posed. Later, a practical level-crossing algorithm that extracts secret bits from channel impulse response is developed in [45]. Further extensions of this technique to wideband systems [46], environments with different variations [47] and multi-antenna systems [48] have also been explored in literature. A comprehensive survey on link-signature based secret key extraction can be found in [49]. For location distinction, link signatures based on the received signal strength [50], the channel gains of multi-tonal probes [51] and the multipath characteristics [52] have been considered in literature, and a comparison of these different forms of link signature is given in [53]. Link-signature based location distinction using MIMO channels has been examined in literature as well [54].

While bringing a new approach to security establishment, link signature naturally exhibits vulnerabilities as many other wireless techniques. Recently, several potential attacks that can severely impair the security established by link-signature based mechanisms have been noticed by researchers. For example, an active virtual multipath attack is proposed in [55] to defeat link-signature based location distinction, in which the attacker proactively creates an “artificial channel” that manifests a multipath propagation feature similar to the real one. In this dissertation, we focus on passive attacks, where the adversary deploys sensors near the legitimate transceivers and aims at inferring the legitimate channel information and the corresponding link signature based on its own channel measurements [56]. To defend against such attacks, guard zones with suitable sizes must be deployed around the legitimate devices. As to this, existing link-signature based security schemes often assume that the legitimate and the adversary channels are essentially uncorrelated as long as the adversary receiver is separated from the legitimate one by more than half a wavelength. A direct implication of this assumption is that a guard zone of half-wavelength is sufficient to protect the legitimate link signature. Although the assumed half-wavelength channel decorrelation has been observed in [57], much larger channel decorrelation distance has been observed in practice as well [58]. These seemingly contradictory facts indicate that channel correlation varies in different environments, which motivates our study on the dependability of link signature in Chapter 8.

In literature, various channel correlation models such as the one-ring model, two-ring model and elliptical scatterer-ring model have been developed for conventional indoor picocell and outdoor micro-/macro-cell communications [59–62]. In Chapter 8, these well-established models are revisited with a focus on exploring the corresponding security implications to link signature. Several key channel factors that have significant influence on link signature security have been identified, and based on the obtained understanding, a generic channel correlation model is developed for link signature security assessment. With this model, suitable guard zone sizes are numerically explored for link-signature based security mechanisms in several typical indoor and outdoor communication scenarios. Moreover, real-world experiments based on Universal Software Radio Peripheral (USRP) platforms and GNURadio are conducted to further justify
the analysis.

### 1.4 Organizations

The remainder of this dissertation is organized as follows. Two countermeasures against the Byzantine attack in CR networks are presented in Chapter 2 and Chapter 3, respectively. The mobile jamming/anti-jamming competition is studied in Chapter 4. The dynamic stochastic security games with information asymmetry are investigated in Chapters 5–7. The security of link signature is examined in Chapter 8. Finally the dissertation is summarized in Chapter 9, together with some directions for future work.
Chapter 2

HMM-Based Malicious User Detection for Robust Collaborative Spectrum Sensing

We start our study from the Byzantine attack in CR networks. As introduced in the previous chapter, if not detected, the Byzantine attackers can severely undermine the normal function of CR networks through sending falsified spectrum sensing results to the fusion center. In this chapter, a new HMM-based malicious user detection method is proposed, where the basic idea is that, by modeling the honest and the malicious users as two HMMs, the HMM parameters determined by users’ sensing behaviors can be exploited for malicious user detection. To this purpose, an appropriate inference technique is required to obtain estimates of the corresponding HMM parameters. It may be tempting to adopt the original Baum-Welch algorithm. However, in the original Baum-Welch algorithm, multiple independent observation sequences are required to estimate a single HMM, and the estimation of multiple HMMs is usually accomplished through using separate training sequences for each individual HMM [63]. Consequently, two specific technical issues prevent the direct application of the original Baum-Welch algorithm to obtain the HMM estimates in the presence of malicious CR users: (1) The local inference sequences reported from secondary users are correlated, because all the secondary users are sensing the same spectrum; (2) Two HMMs exist and the reported sequences from honest users and malicious users are mixed, not separate. To overcome these issues, a new estimation method for HMM is proposed in this chapter. As compared to the original Baum-Welch algorithm, the technical contributions of the proposed algorithm include: 1) simultaneously estimating two HMMs, 2) working with correlated observation sequences and 3) no requirement of separate training sequences, and thus it can be applied to detect malicious users in collaborative spectrum sensing. This method is developed under the generic Expectation-Maximization (EM) framework [64].
Once the estimates of these two HMMs are obtained, two auxiliary tests can be invoked to identify the malicious users by detecting their abnormal spectrum sensing behaviors. Both hard data fusion (where data from detected malicious users is directly discarded) and soft data fusion (where malicious user data are processed and fused along with honest user data) are discussed.

The remainder of this chapter is organized as follows. Section 2.1 formulates the problem. In Section 2.2, explicit expressions of the HMM estimates are derived along with convergence analysis, followed by malicious user detection based on the obtained HMM estimates. Data fusion is discussed in Section 2.3. Simulation results and relevant analysis are given in Section 2.4. Section 2.5 concludes this chapter.

2.1 Problem Formulation

2.1.1 System Model and Spectrum Sensing Metrics

To quantify the spectrum sensing performances, probabilities of detection $P_d$ and false alarm $P_{fa}$ are widely used in literature [4, 65, 66]. In this work, it is assumed that all secondary users use the same devices to sense the spectrum and have the same $\{P_d, P_{fa}\}$. An honest user will directly report its local spectrum state inference to the fusion center, i.e., $P_d^H = P_d$ and $P_{fa}^H = P_{fa}$ from the fusion center’s viewpoint. A malicious user, however, may intentionally tamper its local inferences before reporting to the fusion center. Two parameters $\varphi_{01}$ and $\varphi_{10}$ are used to quantify malicious users’ attacking levels, where $\varphi_{01}$ is the probability that a malicious user will flip its local inference from 0 (idle spectrum) to 1 (occupied spectrum) and $\varphi_{10}$ is the probability of flipping 1 to 0. Accordingly, the equivalent $P_d^M$ and $P_{fa}^M$ of the malicious users to the fusion center are given by

\begin{align*}
P_d^M & = (1 - \varphi_{10})P_d + \varphi_{01}(1 - P_d), \quad (2.1) \\
P_{fa}^M & = (1 - \varphi_{10})P_{fa} + \varphi_{01}(1 - P_{fa}), \quad (2.2)
\end{align*}

respectively. For an effective malicious user, at least one of $\varphi_{01}$ and $\varphi_{10}$ should be nonzero; otherwise, the malicious user will behave identically to honest users in the statistical sense. In addition, in this work, it is assumed that all the malicious users belong to one type, i.e., having the same $\{\varphi_{01}, \varphi_{10}\}$.

\footnote{The extension to the multi-type malicious users case, i.e., different malicious users may have different $\{\varphi_{01}, \varphi_{10}\}$, remains a future work.}
Table 2.1: Notations and symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>Detection window length</td>
</tr>
<tr>
<td>$q_t$</td>
<td>True spectrum state at time $t$</td>
</tr>
<tr>
<td>$q = {q_0, \ldots, q_T}$</td>
<td>Collection of true spectrum states over time</td>
</tr>
<tr>
<td>$S$</td>
<td>Total number of spectrum states</td>
</tr>
<tr>
<td>$L$</td>
<td>Total number of secondary users</td>
</tr>
<tr>
<td>$o^l_t$</td>
<td>The report from the $l$th user at time $t$</td>
</tr>
<tr>
<td>$O^l = {o^l_1, \ldots, o^l_T}$</td>
<td>Report sequence from the $l$th user</td>
</tr>
<tr>
<td>$O_t = {o^1_t, \ldots, o^L_t}$</td>
<td>Reports from all the $L$ users at time $t$</td>
</tr>
<tr>
<td>$O = {O^1, \ldots, O^L}$</td>
<td>All the report sequences from all users</td>
</tr>
<tr>
<td>$N$</td>
<td>Number of user types</td>
</tr>
<tr>
<td>$m^l \in {1, \ldots, N}$</td>
<td>The type of the $l$th user</td>
</tr>
<tr>
<td>$m = {m^1, \ldots, m^L}$</td>
<td>Set of user types of these $L$ users</td>
</tr>
<tr>
<td>$\lambda'$</td>
<td>Estimate of $\lambda$ at the previous iteration</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Percentage of malicious users</td>
</tr>
</tbody>
</table>

Remark: In the interest of space, we will only focus on the above mentioned flipping attack as it is widely considered in literature (e.g., [1, 10, 67, 68]). Nevertheless it is worth noting that besides the flipping attack, malicious users can also modify their operating points ($P^M_d$, $P^M_f$) by changing local detection thresholds [12], and our method can be used to defend against such type of attacks as well.

Figure 2.1: Single user spectrum sensing model.
2.1.2 Two Correlated HMMs

In the context of spectrum sensing, the HMM of a single secondary user is shown in Fig. 2.1, where \( q \) and \( o \) represent the true and the reported spectrum states, respectively. The corresponding set of parameters of this single HMM \( \lambda = \{\pi, A, B\} \) consists of three parts: 1) initial spectrum state distribution vector \( \pi = [\pi_0, \pi_1] \) where \( \pi_i \) is the probability that the spectrum starts from the \( i \)th state; 2) spectrum state transition probability matrix \( A = [a_{i,j}]_{2 \times 2} \) where \( a_{i,j} \triangleq P(q_t = j|q_{t-1} = i) \).\(^2\) For example, \( a_{0,0} \) is the self-transition probability of the idle spectrum state and \( a_{0,1} \) is the transition probability from the idle to the occupied spectrum state; 3) spectrum sensing matrix \( B = [b_{ij}]_{2 \times 2} \) where \( b_{ij} \triangleq b_i(j) = P(o = j|q = i) \) is the probability that the reported spectrum state is \( j \) given that the true spectrum state is \( i \). For example, \( b_1(1) \) is the probability of detection and, similarly, \( b_0(1) \) is the probability of false alarm from the viewpoint of fusion center.

When adversary exists, two HMMs \( \lambda^{(1)}_s = \{\pi^{(1)}, A^{(1)}, B^{(1)}\} \) and \( \lambda^{(2)}_s = \{\pi^{(2)}, A^{(2)}, B^{(2)}\} \) may be used to represent the honest and the malicious users respectively. The abnormal sensing behavior of malicious users causes the difference between \( B^{(1)} \) and \( B^{(2)} \), but the parameters \( \pi \) and \( A \) of these two HMMs are identical as all users are sensing the same spectrum (i.e., user observations are correlated). Specifically, the sensing matrix of the \( \bar{m} \)th (\( m \in \{1, 2\} \) representing honest and malicious users respectively) HMM is denoted by \( B^{(\bar{m})} = \{b_{1\bar{m}}(k)\} \) (e.g., \( b_1^{(1)}(1) \) and \( b_1^{(2)}(1) \) are the probabilities of detection of the CR users corresponding to the first and the second HMM, respectively). Nevertheless, the original Baum-Welch algorithm cannot be applied to estimate these two HMMs as no separate training sequences are available (i.e., honest and malicious data are mixed). Considering this, a new inference algorithm that can jointly estimate these two HMMs based on the mixed and correlated spectrum sensing sequences from all users is developed in this work by introducing a new user classification vector \( c_{1 \times 2} = [c_1, c_2] \). In particular, \( c_m \) (\( m \in \{1, 2\} \)) denotes the percentage of the \( m \)th type of users in the network.

\(^2\)With a slight abuse of notation, the indices of matrices start from 0.
and the entire set of parameters to be estimated is \( \lambda = \{c, \pi, A, B^{(1)}, B^{(2)}\} \). For a fixed \( \lambda \), \( c_{\bar{m}} \) can be interpreted as the prior probability of any user belonging to the \( \bar{m} \)th HMM. To conform with the EM framework, we will express \( c_{\bar{m}} \) as \( P(m^l = \bar{m} | \lambda) \) for any \( l \) in the following derivations, where \( m^l \) denotes the type of the \( l \)th user. After obtaining the estimates of these two HMMs, secondary users can be clustered into two types (each corresponding to one HMM) by evaluating the corresponding maximum a posteriori (MAP) probabilities, and then two auxiliary tests presented in Section 2.2.3 will be invoked to identify the malicious type.

Other notations used in this chapter are summarized in Table 2.1. Specifically, \( O = \{O_1, ..., O^L\} = \{O_1, ..., O_T\} \) is the collection of sensing reports from all these \( L \) users from time \( t = 1 \) to \( t = T \); the number of spectrum states \( S \) equals 2 (i.e., either occupied or idle), and the number of user types \( N \) also equals 2 (i.e., either honest or malicious) in this work. In addition, \( \lambda \) and \( \lambda' \) denote the HMM estimates in the current and the previous iterations, respectively.

### 2.2 The Proposed Method

In this section, the proposed joint estimation method for two HMMs is developed along with the corresponding convergence analysis, and followed by two auxiliary tests for malicious user detection based on these estimates.

#### 2.2.1 Derivations of the Proposed Estimation Method

The Expectation-Maximization algorithm, which alternates between an expectation (E) step and a maximization (M) step, is a generic iterative method for finding maximum likelihood estimates of parameters in a statistical model with hidden variables, such as the HMM. The current E-step formulates the expected log-likelihood function where the distribution of the hidden variables is determined by the optimal parameter \( \lambda' \) found in the last M-step, and current M-step finds the optimal parameter \( \lambda \) that maximizes the expected log-likelihood function formulated in the current E-step [64]. This alternation continues until convergence (or a maximum number of iterations is reached). In this subsection, the E-step and M-step for the estimation of \( \lambda \) based on the collection of all the secondary users’ reported spectrum sensing sequences \( O \) are derived. The collection of the spectrum states \( q \) and the set of user types \( m \) are hidden vectors.

**E-step**

The E-step computes the expectation (with respect to \( q \) and \( m \)) of the log-likelihood function \( \log P(O, q, m | \lambda) \), where the distribution of the hidden variables \( q \) and \( m \) is determined by the

---

3With slight abuse of notation, \( q \) here coincides with the generic one in Fig. 2.1.
estimate $\lambda'$ in the previous iteration and the observed sequences $O$, denoted by $P(q, m|\lambda', O)$. This results in

$$
\tilde{Q}(\lambda, \lambda') = \mathbb{E}\{\log P(O, q, m|\lambda)\} = \sum_{q\in\mathcal{Q}} \sum_{m\in\mathcal{M}} P(q, m|\lambda', O) \log P(O, q, m|\lambda)
$$

$$
= \frac{1}{P(O|\lambda')} \sum_{q\in\mathcal{Q}} \sum_{m\in\mathcal{M}} P(O, q, m|\lambda') \log P(O, q, m|\lambda) \triangleq \frac{1}{P(O|\lambda')} Q(\lambda, \lambda'), \quad (2.3)
$$

where $\mathcal{Q} = \{0, 1\}^T$ and $\mathcal{M} = \{1, 2\}^L$ are the outcome spaces of $q$ and $m$, respectively; $\sum_{q\in\mathcal{Q}}$ and $\sum_{m\in\mathcal{M}}$ are shorthand notations for $\sum_{q_0\in\{0, 1\}} \sum_{q_1\in\{0, 1\}} \cdots \sum_{q_T\in\{0, 1\}}$ and $\sum_{m_1\in\{1, 2\}} \sum_{m_2\in\{1, 2\}} \cdots \sum_{m_L\in\{1, 2\}}$, respectively. In particular, the first equality in (2.3) is due to definition of $\tilde{Q}(\lambda, \lambda')$; the second step expands the expectation with respect to $P(q, m|\lambda', O)$; the Bayes’ formula is applied in the third equality; and the last step is due to the definition of $Q(\lambda, \lambda')$ given below.

It is worth noting that $P(O|\lambda')$ in (2.3) is a constant that does not depend on $\lambda$, which implies that to maximize $\tilde{Q}(\lambda, \lambda')$ is equivalent to maximize

$$
Q(\lambda, \lambda') \triangleq \sum_{q\in\mathcal{Q}} \sum_{m\in\mathcal{M}} P(O, q, m|\lambda') \log P(O, q, m|\lambda).
$$

Therefore, $Q(\lambda, \lambda')$ will be used to find the optimal $\lambda$ in the subsequent derivations without affecting the results.

**M-step**

The M-step aims at finding the current estimate $\lambda$ that maximizes $Q(\lambda, \lambda')$. Note that

$$
P(O, q, m|\lambda) = P(q|\lambda)P(O, m|q, \lambda) = \pi_{q_0} \cdot \prod_{t=1}^{T} a_{q_{t-1}, q_t} \cdot \prod_{l=1}^{L} P(O^l, m^l|q, \lambda)
$$

$$
= \pi_{q_0} \cdot \prod_{t=1}^{T} a_{q_{t-1}, q_t} \cdot \prod_{l=1}^{L} P(O^l|m^l, q, \lambda) \cdot \prod_{l=1}^{L} P(m^l|\lambda), \quad (2.4)
$$

where the second equality holds because all the $L$ reported sequences $O^l$ are conditionally independent given the true spectrum state sequence $q$; the last equality holds because the user type $m^l$ is independent to the true spectrum state $q$; here and onwards $\pi_{q_0}$ represents $\pi_i|_{i=q_0}$, i.e., $\pi_i$ evaluated at $i = q_0$, and $q_0 \in \{0, 1\}$ is the initial spectrum state. According to (2.4),
\[ Q(\lambda, \lambda') \text{ can be expressed as} \]

\[
Q(\lambda, \lambda') = \sum_{q \in Q} \sum_{m \in M} P(O, q, m|\lambda') \log \prod_{l=1}^{L} P(m^l|\lambda) + \sum_{q \in Q} \sum_{m \in M} P(O, q, m|\lambda') \log \pi_{q0} 
\]

\[
+ \sum_{q \in Q} \sum_{m \in M} P(O, q, m|\lambda') \log \prod_{t=1}^{T} a_{qt-1, qt} 
\]

\[
+ \sum_{q \in Q} \sum_{m \in M} P(O, q, m|\lambda') \log \prod_{l=1}^{L} P(O^l|\lambda, q, m^l). \tag{2.5}
\]

Note that in (2.5), each of the four terms \((a), (b), (c), \text{ and } (d)\) relies on only one of the four parameters \(c, \pi, A \text{ and } B\), respectively. This implies that the optimal values of \(c, \pi, A \text{ and } B\), which maximize \(Q(\lambda, \lambda')\), can be found by maximizing these four terms separately. A common approach used in the following derivation is the method of Lagrange multiplier.
Current estimate of $c$ To maximize term $(a)$ in (2.5), it is equivalent to maximize

$$
\sum_{q \in Q} \sum_{m \in M} P(O, q, m|\lambda') \log \prod_{l=1}^{L} P(m^l|\lambda)
$$

$$
= \sum_{m \in M} P(O, m|\lambda') \log \prod_{l=1}^{L} P(m^l|\lambda)
$$

$$
= \sum_{m^1} \sum_{m^2} \cdots \sum_{m^L} P(O, m^1, m^2, \ldots, m^L|\lambda') \sum_{l=1}^{L} \log P(m^l|\lambda)
$$

$$
= \sum_{m^1} \sum_{m^2} \cdots \sum_{m^L} \left[ P(O, m^1, m^2, \ldots, m^L|\lambda') \cdot \sum_{l=1}^{L} \sum_{\bar{m}=1}^{N} \delta(m^l - \bar{m}) \cdot \log P(m^l = \bar{m}|\lambda) \right]
$$

$$
= \sum_{l=1}^{L} \sum_{\bar{m}=1}^{N} \left[ \sum_{m^1} \sum_{m^2} \cdots \sum_{m^L} \delta(m^l - \bar{m}) \cdot P(O, m^1, m^2, \ldots, m^L|\lambda') \cdot \log P(m^l = \bar{m}|\lambda) \right]
$$

$$
= \sum_{l=1}^{L} \sum_{\bar{m}=1}^{N} P(O, m^l = \bar{m}|\lambda') \log P(m^l = \bar{m}|\lambda)
$$

$$
= \sum_{l=1}^{L} \sum_{\bar{m}=1}^{N} P(O, m^l = \bar{m}|\lambda') \log c_{\bar{m}},
$$

(2.6)

where $\delta(n) = 1$ if and only if $n = 0$; the first step is due to marginalization over $q$; the second last step is due to the fact that the expression inside the brackets only depends on $m^l$ and thus all the other $m^{l'}$’s can be marginalized out; the last step is by the definition of $c$.

By associating the constraint $\sum_{\bar{m}=1}^{N} c_{\bar{m}} = 1$ with a Lagrange multiplier $\nu$, the optimal $c$ can be found by solving

$$
\frac{\partial}{\partial c_{\bar{m}}} \left\{ \sum_{l=1}^{L} \sum_{\bar{m}=1}^{N} P(O, m^l = \bar{m}|\lambda') \log c_{\bar{m}} + \nu \left( \sum_{\bar{m}=1}^{N} c_{\bar{m}} - 1 \right) \right\} = 0, \quad \bar{m} = 1, \ldots, N,
$$

which leads to

$$
\sum_{l=1}^{L} \frac{1}{c_{\bar{m}}} P(O, m^l = \bar{m}|\lambda') + \nu = 0, \quad \bar{m} = 1, \ldots, N.
$$

(2.7)
Summing over all \( \bar{m} \) results in \( \nu = -L \cdot P(O | \lambda') \) and

\[
c_{\bar{m}} = \frac{1}{L} \sum_{l=1}^{L} \frac{P(O, m' = \bar{m} | \lambda')}{P(O | \lambda')}, \quad \bar{m} = 1, \ldots, N. \tag{2.8}
\]

**Current estimate of \( \pi \)** By marginalizing over \( m \) and \( q_1, \ldots, q_T \), (b) in (2.5) can be simplified as \( \sum_{i=0}^{S-1} P(O, q_0 = i | \lambda') \log \pi_i \). Using, again, the Lagrange multiplier method, the update equation of \( \pi_i \) based on \( \lambda' \) is given by

\[
\pi_i = \frac{P(O, q_0 = i | \lambda')}{P(O | \lambda')}, \quad i = 0, \ldots, S - 1. \tag{2.9}
\]

**Current estimate of \( A \)** Term (c) in (2.5) can be further simplified as

\[
\sum_{q \in \mathcal{Q}} \sum_{m \in \mathcal{M}} P(O, q, m | \lambda') \log \prod_{t=1}^{T} a_{q_{t-1}, q_t} = \sum_{q \in \mathcal{Q}} P(O, q | \lambda') \sum_{i=0}^{S-1} \sum_{j=0}^{S-1} \delta(q_{t-1} - i) \delta(q_t - j) \log a_{i,j} = \sum_{t=1}^{T} \sum_{i=0}^{S-1} \sum_{j=0}^{S-1} \left[ \sum_{q_1} \ldots \sum_{q_T} P(O, q | \lambda') \cdot \delta(q_{t-1} - i) \delta(q_t - j) \log a_{i,j} \right] = \sum_{t=1}^{T} \sum_{i=0}^{S-1} \sum_{j=0}^{S-1} P(O, q_{t-1} = i, q_t = j | \lambda') \log a_{i,j}, \tag{2.10}
\]

where in the first step marginalization over \( m \) is applied; in the last step all the \( q \)'s other than \( q_{t-1} \) and \( q_t \) are marginalized out because the term inside the brackets only depends on \( q_{t-1} \) and \( q_t \). Using the same method, the update equation of \( a_{i,j} \) based on \( \lambda' \) can be found as

\[
a_{i,j} = \frac{\sum_{t=1}^{T} P(O, q_{t-1} = i, q_t = j | \lambda')}{\sum_{t=1}^{T} P(O, q_{t-1} = i | \lambda')}, \quad i, j = 0, \ldots, S - 1. \tag{2.11}
\]
Current estimate of $B$ Term (d) in (2.5) can be further simplified as

$$\sum_{q \in Q} \sum_{m \in M} P(O, q, m | \lambda') \log \prod_{l=1}^{L} P(O^l | \lambda, q, m^l)$$

$$= \sum_{q \in Q} \sum_{m \in M} P(O, q, m | \lambda') \sum_{l=1}^{L} \sum_{t=1}^{T} \log P(o^t_l | \lambda, q_t = i, m^l = \bar{m})$$

$$= \sum_{l=1}^{L} \sum_{t=1}^{T} \sum_{i=0}^{S-1} \sum_{\bar{m}=1}^{N} \left( P(O, q_t = i, m^l = \bar{m} | \lambda') \cdot \log \sum_{k=0}^{S-1} \delta(o^t_l - k)b^m_{i}(k) \right) \tag{2.12}$$

where $P(O^l | \lambda, q, m^l) = \prod_{t=1}^{T} P(o^t_l | \lambda, q_t, m^l)$ is applied in the first step, i.e., the observations $o^t_l$'s are mutually conditional independent and $o^t_l$ only depends on $q_t$ given $\lambda$ and the user type $m^l$; by definition $b^m_{i}(o^t_l) = P(o^t_l | \lambda, q_t = i, m^l = \bar{m})$ and thus the last step holds. Similarly, the update equation of $b^m_{i}(k)$ based on $\lambda'$ is given by

$$b^m_{i}(k) = \frac{\sum_{l=1}^{L} \sum_{t=1}^{T} \delta(o^t_l - k)P(O, q_t = i, m^l = \bar{m} | \lambda')}{\sum_{l=1}^{L} \sum_{t=1}^{T} P(O, q_t = i, m^l = \bar{m} | \lambda')} \tag{2.13}$$

$$i = 0, ..., S - 1, \; \bar{m} = 1, ..., N \text{ and } k = 0, ..., S - 1.$$

2.2.2 Convergence Analysis

The convergence of the log-likelihood $\{ \log P(O | \lambda_n) \}$ of the proposed algorithm is shown in this subsection, where $\lambda_n$ is the estimate of $\lambda$ obtained in the $n$th EM iteration. For this purpose, define for any $\bar{\lambda}$

$$\Delta(\bar{\lambda} | \lambda_n) \triangleq \sum_{q \in Q} \sum_{m \in M} P(q, m | \lambda_n) \log \frac{P(O, q, m | \bar{\lambda})P(q, m | \lambda)}{P(q, m | O, \lambda_n)P(O | \lambda_n)} \tag{2.14}$$
Φ(\bar{\lambda}|\lambda_n) \triangleq \log P(O|\lambda_n) + \Delta(\bar{\lambda}|\lambda_n) \\
= \sum_{q \in Q} \sum_{m \in M} P(q, m|O, \lambda_n) \log \frac{P(O|q, m, \bar{\lambda})P(q, m|\bar{\lambda})}{P(q, m|O, \lambda_n)} \\
= \tilde{Q}(\bar{\lambda}, \lambda_n) + \text{const. w.r.t. } \bar{\lambda}.

(2.15)

First, observe that \lambda_{n+1} found in the \((n+1)\)th M-step maximizes \(Q(\bar{\lambda}, \lambda_n) \propto \tilde{Q}(\bar{\lambda}, \lambda_n)\), and thus it maximizes \(\Phi(\bar{\lambda}|\lambda_n)\) given in (2.15), which implies \(\Phi(\lambda_{n+1}|\lambda_n) \geq \Phi(\lambda_n|\lambda_n)\). In addition, it is shown in Appendix A.1 that \(\log P(O|\bar{\lambda}) \geq \Phi(\bar{\lambda}|\lambda_n)\) \(\forall \bar{\lambda} \) and \(\log P(O|\lambda_n) = \Phi(\lambda_n|\lambda_n)\). Therefore, by setting \(\bar{\lambda} = \lambda_{n+1}\), it follows that \(\log P(O|\lambda_{n+1}) \geq \log P(O|\lambda_n)\) \(\forall n\), i.e., \(\{\log P(O|\lambda_n)\}\) is a non-decreasing sequence of \(n\). Then, the convergence of \(\{\log P(O|\lambda_n)\}\) follows by further observing that it is bounded above by 0.

2.2.3 Implementation

The calculation of update equations (2.8), (2.9), (2.11) and (2.13) requires evaluation of the corresponding probability quantities presented in the numerators and denominators (e.g., \(P(O, m|q, \lambda')\) and \(P(O|\lambda')\) in (2.8)). To this end, similar to the original Baum-Welch algorithm, intermediate variables are defined. Firstly, define

\[\alpha_{i}^{l,m}(t) = P(O_{1}, O_{2}, ..., O_{t}, q_{t} = i|\lambda', m^{l} = \bar{m}),\]

which is the probability of the fusion center observing the partial sequence \(\{O_{1}, O_{2}, ..., O_{t}\}\) and the spectrum ending up in state \(i\) at time \(t\) given that the type of the \(l\)th user \(m^{l}\) is \(\bar{m}\). Secondly, define

\[\beta_{i}^{l,m}(t) = P(O_{t}, O_{t+1}, ..., O_{T}|q_{t} = i, \lambda', m^{l} = \bar{m}),\]

which is the probability of the fusion center seeing partial sequence \(\{O_{t}, O_{t+1}, ..., O_{T}\}\) given that the spectrum started at state \(i\) at time \(t\) and the \(l\)th user is of type \(\bar{m}\). Further define

\[\xi_{ij}^{l,m}(t) \triangleq P(O, q_{t} = i, q_{t+1} = j|\lambda', m^{l} = \bar{m}),\]

and

\[\gamma_{i}^{l,m}(t) \triangleq P(O, q_{t} = i|\lambda', m^{l} = \bar{m}),\]

which admit \(\xi_{ij}^{l,m}(t) = \sum_{j=1}^{S-1} \xi_{ij}^{l,m}(t)\) by definition.

Once \(\alpha, \beta, \gamma\) and \(\xi\) are obtained, it can be verified (using marginalization and the Bayes’
formula) that (2.8), (2.9), (2.11), (2.13) can be evaluated as follows (see Appendix A.2):

\[ c_{\bar{m}} \propto \frac{1}{T} \sum_{l=1}^{L} \left( c'_{\bar{m}} \sum_{i=0}^{S-1} \alpha_{i}^{l,\bar{m}}(T) \right), \quad \bar{m} = 1, \ldots, N, \quad (2.16) \]

\[ \pi_{i} \propto \sum_{\bar{m}=1}^{N} \left( \alpha_{i}^{l,\bar{m}}(0)c_{\bar{m}} \sum_{j=0}^{S-1} \beta_{j}^{l,\bar{m}}(1) \right), \quad i = 0, \ldots, S - 1, \quad (2.17) \]

\[ a_{i,j} = \frac{\sum_{t=1}^{T} \sum_{\bar{m}=1}^{N} \xi_{i,j}^{l,\bar{m}}(t-1)c'_{\bar{m}}}{\sum_{j=0}^{S-1} \sum_{t=1}^{T} \sum_{\bar{m}=1}^{N} \xi_{i,j}^{l,\bar{m}}(t-1)c'_{\bar{m}}}, \quad i, j = 0, \ldots, S - 1, \quad (2.18) \]

\[ b_{i}^{m}(k) = \frac{\sum_{l=1}^{L} \sum_{t=1}^{T} \delta(\sigma_{i}^{l} - k)\gamma_{i}^{l,\bar{m}}(t)c'_{\bar{m}}}{\sum_{l=1}^{L} \sum_{t=1}^{T} \gamma_{i}^{l,\bar{m}}(t)c'_{\bar{m}}}, \quad i = 0, \ldots, S - 1, \quad \bar{m} = 1, \ldots, N \text{ and } k = 0, \ldots, S - 1, \quad (2.19) \]

where \( c'_{\bar{m}} \) denotes the estimated percentage of type \( \bar{m} \) users from the last iteration, i.e., \( c'_{\bar{m}} = P(m^{l} = \bar{m} | \lambda') \).

Precise evaluation of \( \alpha, \beta \) and \( \xi \) are computationally intractable, and so is \( \gamma \). To reduce the computation complexity, three approximations are adopted in this work:

\((A1)\) \( P(O_{t+1}|q_{t+1} = j, O_{1}, O_{2}, \ldots, O_{t}, q_{t} = i, \lambda', m^{l} = \bar{m}) \approx P(O_{t+1}|q_{t+1} = j, q_{t} = i, \lambda', m^{l} = \bar{m}) \),

\((A2)\) \( P(O_{t+1}, \ldots, O_{T}|q_{t+1} = j, q_{t} = i, O_{t}, \lambda', m^{l} = \bar{m}) \approx P(O_{t+1}, \ldots, O_{T}|q_{t+1} = j, q_{t} = i, \lambda', m^{l} = \bar{m}) \),

\((A3)\) \( P(O_{t+1}, \ldots, O_{T}|q_{t+1} = j, O_{1}, \ldots, O_{t}, q_{t} = i, \lambda', m^{l} = \bar{m}) \approx P(O_{t+1}, \ldots, O_{T}|q_{t+1} = j, \lambda', m^{l} = \bar{m}) \).

The intuition of these approximations is the following. Note that if there is only one type of users (i.e., single HMM), the future spectrum sensing results \( \{O_{t+1}, \ldots, O_{T}\} \) are independent of previous sensing results \( \{O_{1}, \ldots, O_{t}\} \) given the true spectrum state \( q_{t+1} \) due to the Markovian
property. Consequently, the left hand side will equal to the right hand side in each of the above three expressions. However, when there are more than one type of users, the probability of observing \(\{O_{t+1}, \ldots, O_T\}\) (and \(O_{t+1}\)) depends on not only the true spectrum states \(q\) but also the user types \(m\). In the above expressions, user types (other than the \(l\)th one \(m_l\)) are not known explicitly but contained in the previous observations \(\{O_1, \ldots, O_t\}\) (as previous observations can be used to estimate user types). Therefore, the probability of observing \(\{O_{t+1}, \ldots, O_T\}\) (and \(O_{t+1}\)) will be affected by \(\{O_1, \ldots, O_t\}\). (For single HMM, user types are always known, and thus it is free from this problem.) This makes the precise computations of \(\alpha\), \(\beta\) and \(\xi\) intractable.

Considering this, the user type information contained in previous sensing results is discarded so as to reduce the computation complexity to a tractable level. In particular, \(\{O_1, O_2, \ldots, O_t\}\) is discarded in (A1) and (A3), and \(O_t\) is discarded in (A2), respectively. Then, the following formulas can be used to evaluate \(\alpha\), \(\beta\) and \(\xi\) iteratively (see Appendix A.3):

\[
\alpha_{l,m}(t+1) \approx \sum_{i=0}^{S-1} a_{i,j} b_{j}^{m}(O_{t+1}) \prod_{\bar{m}=1, \bar{m} \neq m}^{N} b_{j}^{\bar{m}}(O_{t+1})c_{\bar{m}}^{m}, \tag{2.20}
\]

\[
\beta_{l,i}(t) \approx \left[ b_{1}^{m}(O_{t+1}) \prod_{\bar{m}=1, \bar{m} \neq m}^{N} b_{j}^{\bar{m}}(O_{t+1})c_{\bar{m}}^{m} \right] \sum_{j=0}^{S-1} a_{i,j} b_{j}^{m}(O_{t+1}) \xi_{ij}(t+1), \tag{2.21}
\]

where \(\alpha_{l,m}(0) \equiv \pi_{i}\) and \(\beta_{l,i}(T+1) \equiv 1\), and

\[
\xi_{ij}(t) \approx \alpha_{i,j} b_{j}^{m}(O_{t+1}) \beta_{l,j}(t+1). \tag{2.22}
\]

The approximate values of \(\gamma\) can be calculated according to \(\gamma_{l,i}(t) = \sum_{j=1}^{S-1} \xi_{ij}(t)\).

Remark: Since all the users sense the same spectrum, if \(\alpha\), \(\beta\) and \(\xi\) can be computed precisely, (2.17) and (2.18) will result in the same update to \(\pi_i\) and \(a_{i,j}\), respectively, for any user ID \(l\). However, due to the approximations (A1)–(A3), this property does not hold. Nevertheless the simulations show that using any (fixed) \(l\) in the evaluations of (2.17) and (2.18) results in similar estimation performance.

### 2.2.4 Secondary User Classification

Once the estimate \(\hat{\lambda}\) is obtained, the next step is to classify secondary users, which consists of two stages: 1) dividing all the secondary users into two groups and 2) identifying the malicious group.

---

*In (2.20), (2.21) and (2.22), \(a_{i,j}\), \(b_{j}^{m}\), \(\pi_i\) and \(c_{\bar{m}}^{m}\) are the estimates in \(\lambda'\).*
Stage I: Based on the estimate $\hat{\lambda}$ obtained after the EM algorithm converges\(^5\) and the collection of observation sequences $O$, all secondary users are divided into two groups (each corresponding to one HMM) and the a posteriori that the $l$th user belongs to the $\bar{m}$th group is given by

$$P(m^l = \bar{m} | \hat{\lambda}, O) = \frac{P(m^l = \bar{m} | \hat{\lambda}) P(O | m^l = \bar{m}, \hat{\lambda})}{P(O | \hat{\lambda})} = \left( \hat{c}_{\bar{m}} \sum_{i=0}^{S-1} \alpha_{i}^{l,\bar{m}}(T) \right) / \sum_{\bar{m}=1}^{N} \left( \hat{c}_{\bar{m}} \sum_{i=0}^{S-1} \alpha_{i}^{l,\bar{m}}(T) \right).$$

(2.23)

Therefore, the MAP estimate of the associated group of the $l$th user is determined by

$$\hat{m}^l = \arg \max_{\bar{m}} \hat{c}_{\bar{m}} \sum_{i=0}^{S-1} \alpha_{i}^{l,\bar{m}}(T) / \sum_{\bar{m}=1}^{N} \left( \hat{c}_{\bar{m}} \sum_{i=0}^{S-1} \alpha_{i}^{l,\bar{m}}(T) \right).$$

(2.24)

Stage II: After dividing all secondary users into two groups, two tests are proposed to identify the malicious group: the spectrum sensing ability (SSA) test and group size (GS) test. Specifically, the SSA of a sensing matrix $B$ is defined as $SSA(B) \triangleq |b_{11} - b_{01}| = |P_d - P_{fa}|$. According to (2.1) and (2.2), it is proved in Appendix A.4 that the SSA of the malicious user’s sensing matrix is always no greater than that of the honest user, i.e., $SSA(B^M) \leq SSA(B^H)$. The intuition is that malicious users cannot increase their spectrum sensing ability through data processing. Consequently, the group of users with lower SSA are identified as malicious. In the GS test, the group with the smaller number of users will be identified as malicious.

Since the estimate of $B$ may not be perfect in practice, a threshold $SSA_{th}$ is set such that only when the difference between the two SSAs is larger than $SSA_{th}$ will the SSA test be used.\(^6\) However, when the attacking level $\varphi$ of the malicious user is close to $\{1, 1\}$, the difference between $SSA(B^M)$ and $SSA(B^H)$ is close to zero and the SSA test will fail to detect the malicious users.\(^7\) Fortunately, the grouping will be very accurate in such cases, due to the significant difference in statistical behaviors between the honest and malicious users. Consequently, the GS test can be used (when the percentage of malicious users is less than half).\(^8\) In particular, when $|SSA(\hat{B}^{(1)}) - SSA(\hat{B}^{(2)})| \leq SSA_{th}$, the GS test will be activated.

\(^5\)Usually, the estimates of the proposed algorithm converge to reasonably accurate values in less than 30 iterations.

\(^6\)Note that the proposed HMM-based algorithm divides the secondary users into two clusters without involving any threshold; here, the threshold $SSA_{th}$ is only used to identify which cluster is malicious.

\(^7\)Malicious user detection for $\varphi$ close to $\{0, 0\}$ is not of our focus because in such cases malicious users behave nearly identically to honest users in the statistical sense.

\(^8\)To relax the assumption that the percentage of malicious users is less than half, one possible way in practice is to allocate a few anchor nodes in the network to aid honest group identification, but this is beyond the scope of this work.
2.3 Hard- and Soft Data Fusion

In this section, we will discuss how the fusion center processes the sensing reports based on the detection results. In particular, two different data fusion approaches, which differ in how the fusion center disposes of the data from detected malicious users, are discussed.

**Hard data fusion:** With the user type estimates \( \hat{m} \) given in (2.24), a straightforward (yet effective, as shown by numerical results) strategy of the fusion center is to directly discard all the local spectrum inferences from the detected malicious users, and apply majority voting on the data from honest users to decide the spectrum states. The block diagram of this approach is illustrated in Fig. 2.3. The fusion center maintains a data buffer for the most recent sensing reports (of length \( T \)) from all secondary users, and adopts the proposed algorithm to estimate the corresponding HMM parameters. The resulting estimate \( \hat{\lambda} \) together with the observation history \( O_{t-T:t-1} \) is fed into the MAP block for malicious user identification using (2.24). Spectrum occupancy decision is made at the data fusion block, based on the current sensing reports \( O_t \) and user classification result \( \hat{m} \).

**Soft data fusion:** It is interesting to note that when precise knowledge of HMM parameters \( \lambda \) and user type information \( m \) is available, malicious users may also provide useful information on the spectrum states. Soft data fusion will exploit such potential information in the hope of further enhancing the performance. To this end, the estimates \( \hat{\lambda} \) and \( \hat{m} \) obtained by the proposed inference method can be fed into the data fusion block in Fig. 2.3. Accordingly, the log-likelihood \( L_i(t) \) of the true spectrum state at time \( t \) being \( q(t) = i \), based on the estimates of HMM parameters \( \hat{\lambda} \) and user type \( \hat{m} \), is given by

\[
L_i(t) \triangleq \log P(o_t^1, o_t^2, ..., o_t^L | q(t) = i, \hat{\lambda}, \hat{m}) = \sum_{l=1}^{L} \log \{ o_t^l | q(t) = i, \hat{\lambda}, \hat{m} \} = \sum_{l=1}^{L} \log \hat{b}_{i}^l \hat{m}_l(o_t^l), \quad i = 0, 1.
\]

(2.25)

The log-likelihood ratio is defined as \( \zeta(t) \triangleq L_1(t) - L_0(t) \), based on which the fusion center
Table 2.2: Comparison of $\hat{e}(\hat{B})$ using different $l$’s in (2.17) and (2.18).

<table>
<thead>
<tr>
<th>$\hat{e}(\hat{B})$</th>
<th>$\rho = 45%$</th>
<th>$\rho = 15%$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$l = 1$</td>
<td>0.0854</td>
<td>0.0596</td>
</tr>
<tr>
<td>Random $l$</td>
<td>0.0849</td>
<td>0.0596</td>
</tr>
<tr>
<td>Average over all $l$</td>
<td>0.0827</td>
<td>0.0595</td>
</tr>
</tbody>
</table>

will decide the spectrum state as

$$\hat{q}(t) = \begin{cases} 1, & \zeta(t) > 0, \\ 0, & \text{otherwise.} \end{cases} \quad (2.26)$$

Intuitively, a performance gain of soft data fusion over hard data fusion is expected for precise $\lambda$ and $m$, as additional information is exploited. Performances of hard- and soft data fusions using estimated $\lambda$ and $m$ will be explored numerically in Section 2.4.3.

2.4 Simulations

The efficacy of the proposed method is explored through two measures: 1) the honest/malicious user classification accuracy ($P_{clsf}^m$ and $P_{clsf}^f$) and 2) the spectrum sensing performance at the fusion center ($P_{FC}^d$ and $P_{FC}^f$). Specifically the performances of three fusion centers adopting the majority voting rule are compared; FC1 employs our proposed method that processes the sensing reports from all users simultaneously for malicious user detection and removes detected malicious data from consideration; FC2 first uses the original Baum-Welch algorithm to estimate $\lambda_s$ for each individual user and then employs an agglomerative clustering method in [69], with which users are classified based on the similarity of corresponding $\lambda_s$, for malicious user detection; FC3 does not employ any malicious user detection mechanism. Throughout the simulations, the initial values of $\lambda = \{c, \pi, A, B\}$ are given by $c_{init} = [0.5, 0.5]$, $\pi_{init} = [0.5, 0.5]$, $A_{init} = [0.5, 0.5]$, $B^{(1)}_{init} = [0.8, 0.2]$ and $B^{(2)}_{init} = [0.5, 0.5]$. The detection window length $T$ is 100(time slot). The selection of $SSA_{th}$ depends on the spectrum sensing ability of honest users $SSA(B^H)$. Specifically, it may be chosen as $SSA_{th} = \kappa \cdot SSA(B^H)$ ($\kappa \in (0, 1)$). However, it is found that the performance is not very sensitive to the specific value of $SSA_{th}$. Thus, in the following simulations, $SSA_{th}$ is set to 0.1 for simplicity.

2.4.1 A Basic Example

A basic example is shown first to demonstrate the effectiveness of the proposed method in both detecting malicious users and improving the collaborative spectrum sensing performance. In
Table 2.3: Comparison of HMM parameter estimation errors.

<table>
<thead>
<tr>
<th>$e(\lambda)$</th>
<th>$\rho = 20%$</th>
<th>$\rho = 40%$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P^H_d = 0.8$, $P^H_f = 0.2$</td>
<td>0.0493</td>
<td>0.0701</td>
</tr>
<tr>
<td>$P^H_d = 0.9$, $P^H_f = 0.1$</td>
<td>0.0351</td>
<td>0.0475</td>
</tr>
</tbody>
</table>

Table 2.4: Performance comparison for the three fusion centers.

<table>
<thead>
<tr>
<th></th>
<th>Classification</th>
<th>$P_{FC}^d$</th>
<th>$P_{FC}^f_a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>FC1</td>
<td>99.3%</td>
<td>0.9968</td>
<td>0.0046</td>
</tr>
<tr>
<td>FC2</td>
<td>66.5%</td>
<td>0.8832</td>
<td>0.0661</td>
</tr>
<tr>
<td>FC3</td>
<td></td>
<td>0.7895</td>
<td>0.0632</td>
</tr>
</tbody>
</table>

Figure 2.4: Comparison of spectrum sensing performances over 100 Monte Carlo runs.
this example, \( \rho = 45\% \) of the 20 secondary users are malicious. The true spectrum sensing matrix \( B \) of the honest users is \( \begin{bmatrix} 0.8 & 0.2 \\ 0.15 & 0.85 \end{bmatrix} \) (i.e., \( P_H^d = 0.85 \) and \( P_H^f = 0.2 \)). For malicious users, the attacking level indices are \( \varphi_{10} = 0.8 \) and \( \varphi_{01} = 0.75 \) resulting in a spectrum sensing matrix \( \begin{bmatrix} 0.36 & 0.64 \\ 0.72 & 0.28 \end{bmatrix} \) according to (2.1) and (2.2). The true spectrum state transition matrix \( A \) is set with \( a_{0,1} = a_{1,0} = 0.2 \). 100 Monte Carlo runs are implemented for this scenario. Recall that using different user ID \( l \)'s to evaluate (2.17) and (2.18) may result in different estimates due to the three approximations mentioned in Section 2.2.3. Since the proposed malicious user detection method mainly relies on the estimates of sensing matrices \( B \)'s, the sensing matrices estimation accuracy of using different \( l \)'s is shown in Table 4.1, where the estimation error is defined as 
\[
e(\hat{B}) \triangleq \frac{1}{N_S^2} \sum_{m=1}^{N} \sum_{i=1}^{S} \sum_{j=1}^{S} |\hat{b}_{ij}^m - b_{ij}^m|.
\]
As shown in Table 4.1, using different \( l \)'s leads to similar estimation accuracy for sensing matrices \( B \)'s (more uniform for smaller \( \rho \)).

In the following simulations, \( l = 1 \) is used in the evaluation of (2.17) and (2.18). Furthermore, the HMM parameter estimation errors \( e(\hat{\lambda}) \) (defined similarly as \( e(\hat{B}) \)) are shown in Table 3.2. As it can be seen, the estimation accuracy is acceptable and more accurate estimates can be obtained with better honest sensing devices.

The proposed algorithm achieves high classification accuracy as shown in Table 3.1, where the average classification accuracy is 99.3\%, which outperforms the baseline FC2 whose classification accuracy (66.5\%) is far below a satisfactory level. As a consequence, with FC1, \( P_{d}^{FC} \) is increased from 0.8832 (0.7895) to 0.9968 and \( P_{fa}^{FC} \) is reduced from 0.0661 (0.0632) to 0.0046, as compared to FC2 (FC3). Further, Fig. 2.4 compares these three fusion centers’ spectrum sensing performances, where the \( P_{d}^{FC} \) and \( P_{fa}^{FC} \) of FC1 consistently outperform those of FC2 and FC3. It is also observed in simulations that the proposed method is faster than the baseline approach FC2, as in the proposed method the estimation is done in an integrated manner, with more efficient use of the data and much less redundancy in computation.

### 2.4.2 Further Simulations

To provide a more concrete evaluation of the proposed method, the effects of different malicious user percentage \( \rho \) and attacking level \( \varphi \) are investigated.\(^{11}\) For every pair of \( \rho \) and \( \varphi \), 100 Monte Carlo runs are implemented. Fig. 2.5 shows the performance comparison of all three fusion centers with \( L = 20 \).

Fig. 2.5a shows the regions of \( (\rho, \varphi) \) (under the curves) when both the mis-detection probability \( P_{m}^{dfs} \) and false alarm probability \( P_{fa}^{dfs} \) of malicious user detection are less than 5\% for FC1 and FC2, respectively. The \( P_{m}^{dfs} \) and \( P_{fa}^{dfs} \) of FC2 are always greater than 15\% when

---

9 Similar trends are observed for other parameters \( c, \pi \) and \( A \).
10 About 3 times based on our current implementation.
11 It is assumed in this subsection that \( \varphi_{01} = \varphi_{10} = \varphi \) for 2-D plotting.
and Figure 2.5: Spectrum sensing and classification performances comparisons (SSA to 1, the difference between spectrum sensing performances are similar for two different thresholds, data fusion process. Furthermore, as shown in Fig. 2.5, both malicious user detection and spectrum sensing performances are compared to FC3 can be explained by its poor classification results, as shown in Fig. 2.5a, that effectively without activating the GS test. In addition, the performance degradation of FC2 as age of malicious users exceeds 50% as shown in Fig. 2.5. The reason is that when accurately user classification of the proposed method, the spectrum sensing performance of FC1 is enhanced as compared to the other two fusion centers. The regions where $P_{m}^{\text{clsf}}, P_{fa}^{\text{clsf}} \leq 5\%$. Both $P_{m}^{\text{clsf}}$ and $P_{fa}^{\text{clsf}}$ are small when the inference is accurate.)

Due to accurate user classification of the proposed method, the spectrum sensing performance of FC1 is significantly outperforms FC2. (Note that in the proposed algorithm, there is no tradeoff between $P_{m}^{\text{clsf}}$ and $P_{fa}^{\text{clsf}}$ with respect to $\rho$ and $\varphi$. Both $P_{m}^{\text{clsf}}$ and $P_{fa}^{\text{clsf}}$ are small when the inference is accurate.)

$\varphi \geq 0.3$ resulting in a vanished region in Fig. 2.5a, while FC1 using the proposed method significantly outperforms FC2. (Note that in the proposed algorithm, there is no tradeoff between $P_{m}^{\text{clsf}}$ and $P_{fa}^{\text{clsf}}$ with respect to $\rho$ and $\varphi$. Both $P_{m}^{\text{clsf}}$ and $P_{fa}^{\text{clsf}}$ are small when the inference is accurate.)

Figure 2.5: Spectrum sensing and classification performances comparisons ($L = 20, P_{d}^{H} = 0.85$ and $P_{fa}^{H} = 0.15$). The shaded region corresponds to the method in [1].

\[ \varphi \geq 0.3 \text{ resulting in a vanished region in Fig. 2.5a, while FC1 using the proposed method significantly outperforms FC2. (Note that in the proposed algorithm, there is no tradeoff between } P_{m}^{\text{clsf}}, P_{fa}^{\text{clsf}} \text{ with respect to } \rho \text{ and } \varphi. \text{ Both } P_{m}^{\text{clsf}} \text{ and } P_{fa}^{\text{clsf}} \text{ are small when the inference is accurate.)} \]

\[ \varphi \geq 0.3 \text{ resulting in a vanished region in Fig. 2.5a, while FC1 using the proposed method significantly outperforms FC2. (Note that in the proposed algorithm, there is no tradeoff between } P_{m}^{\text{clsf}}, P_{fa}^{\text{clsf}} \text{ with respect to } \rho \text{ and } \varphi. \text{ Both } P_{m}^{\text{clsf}} \text{ and } P_{fa}^{\text{clsf}} \text{ are small when the inference is accurate.)} \]
classification and spectrum sensing performances are nearly the same for \( \kappa \in [0.1, 0.3] \). This verifies our previous claim and selection of the SSA\(_{th} \) value. It is also worth noting that while the proposed method only provides incremental performance improvement as compared to the method in [1] (see the shaded regions in Fig. 2.5), it is non-reputation based, and can infer two HMMs simultaneously using mixed and correlated data, which may assume wider applications beyond malicious user detection.\(^{12}\)

Malicious user detection performance of the proposed method for CR networks with different number of users \( L \) are compared in Fig. 2.6. As shown, the good performance of the proposed method is (nearly) invariant to the number of users. For example, even when \( \rho = 40\% \) users are malicious and the attacking level \( \varphi \) is as high as 0.9, \( P_{\text{clsf}}^{\text{m}} \) and \( P_{\text{clsf}}^{\text{fa}} \) of the proposed method are always less than 5% for different number of users. As a result of accurate malicious user detection, similar trends of \( P_{\text{FC}}^{\text{d}} \) and \( P_{\text{FC}}^{\text{fa}} \) are observed for different \( L \)'s (figures are omitted).

### 2.4.3 Comparison of Hard- and Soft Data Fusion

Previous results of \( P_{\text{FC}}^{\text{d}} \) and \( P_{\text{FC}}^{\text{fa}} \) are based on hard data fusion. As discussed in Section 2.3, soft data fusion may help to improve spectrum state inference at the fusion center. Its performance is compared with that of hard data fusion in Fig. 2.7–2.8.

The comparison in Fig. 2.7 assumes precise knowledge of the HMM parameters \( \lambda \) and user type information \( m \). For different values of \((P_{\text{d}}, P_{\text{fa}})\)'s, soft data fusion always enables the fusion center to achieve \( P_{\text{d}}^{\text{FC}} \geq 0.95 \) and \( P_{\text{fa}}^{\text{FC}} \leq 0.05 \) and the curves of corresponding operational regions overlap on the top, as shown in Fig. 2.7a and Fig. 2.7b, respectively. Although the operational region of hard data fusion is enhanced with better \( P_{\text{d}} \) and \( P_{\text{fa}} \), it is outperformed by soft data fusion when the number of available honest users for data fusion is small (i.e., when \( \rho \) is large). In contrast, the results in Fig. 2.8 are based on estimated \( \hat{\lambda} \) and \( \hat{m} \). As it can be seen, for relatively poor device sensing capability (\( P_{\text{d}} = 0.75 \) and \( P_{\text{fa}} = 0.25 \)), the performance of soft data fusion is even worse. As device sensing capability increases (e.g., \( P_{\text{d}} = 0.95 \) and \( P_{\text{fa}} = 0.05 \)), soft data fusion only provides comparable performance as to the hard data fusion. One possible explanation is that, with relatively poor devices, the errors introduced in the estimates will deteriorate the soft data fusion, while with good-quality devices, hard data fusion already provides excellent performance. There seems to be no favorable regions for soft data fusion with real estimates based on our simulations.

\(^{12}\)For example, in multi-target tracking [70–76], target classification is an important issue and the proposed algorithm may be applied with some proper variations.
Figure 2.6: Malicious user detection performance for different $L$ ($SSA_{th} = 0.1$, $P_d^H = 0.85$ and $P_{fa}^H = 0.15$).

Figure 2.7: Hard- and soft data fusion comparison with perfect $\hat{\lambda}$ and $\hat{m}$ ($L = 20$).

Figure 2.8: Hard- and soft data fusion comparison with estimated $\hat{\lambda}$ and $\hat{m}$ ($L = 20$).
2.5 Conclusions

An effective malicious user detection method based on the hidden Markov model (HMM) was proposed in this chapter. In particular, by introducing a new user classification vector, the proposed method can estimate two HMMs jointly based on mixed and correlated spectrum observation sequences from secondary users. In addition, two effective tests were proposed to identify the malicious users based on the obtained HMMs estimates. Simulation results show that the proposed method is capable of providing high malicious user detection accuracy and leads to more robust and reliable collaborative spectrum sensing performance at the fusion center in the presence of malicious users, as compared to the baseline approaches. Further, the potential performance gain of soft data fusion over hard data fusion is explored.
Chapter 3

A Byzantine Attack Defender in Cognitive Radio Networks: the Conditional Frequency Check

In this chapter, another method of defending against the Byzantine attack, termed Conditional Frequency Check (CFC), is proposed, which further improves the detection accuracy as compared to the HMM-based method developed in Chapter 2. In particular, two natural but effective CFC statistics, which explore the second order property of the assumed Markov spectrum model, are constructed. To eliminate the requirement of prior information on sensing and spectrum modeling, two consistent histogram estimators based on the history of sensor reports are also developed for these two CFC statistics. In addition, the concept of detection margin (DM) is introduced to quantify the detectability of malicious sensors. To further strengthen the malicious sensor detection capability of the proposed CFC, an auxiliary hamming distance check (HDC) is applied subsequently. With the aid of one trusted sensor and a sufficient long detection window, the proposed CFC can detect any malicious sensor regardless of the proportion of malicious ones in the network.\(^1\) In the case when such a trusted sensor is not available, an auxiliary clustering procedure is developed, which can effectively detect the malicious sensors as long as honest sensors dominate the network.

The rest of this chapter is organized as follows. Section 3.1 formulates the problem. The proposed malicious sensor detection method and the corresponding analytical results are presented in Section 3.2. Threshold selection is discussed in Section 3.3. Extensions of the proposed methods are discussed in Section 3.4. Simulations and corresponding discussion are presented in Section 3.5, and Section 3.6 concludes this chapter.

\(^1\)The assumption of one available trusted sensor has been adopted in literature (e.g., [15]); for instance, the common access point or a sensor itself [14] can serve this role in distributed cooperative spectrum sensing.
3.1 Problem Formulation

In this chapter, the following scenario is considered: 1) The true spectrum has two states, i.e., 0 (idle) and 1 (occupied), and follows a homogeneous Markov model with state transition matrix \( A = [a_{ij}]_{2 \times 2} \) where \( a_{ij} \triangleq Pr(s_{t+1} = j | s_t = i) \) and \( s_t \) denotes the true spectrum state at time \( t \). The stationary spectrum state distribution is denoted by \( \pi = [\pi_0, \pi_1] \), which satisfies \( \pi A = \pi \). In addition, it is assumed that the Markov chain of spectrum states is in equilibrium. 2) There are two types of sensors, honest and malicious ones, and the corresponding sets are denoted by \( H \) and \( M \), respectively. One trusted honest sensor exists and is known by the fusion center. 3) The probabilities of detection and false alarm for spectrum sensing of honest sensors are assumed the same, and denoted by \( P_d \) and \( P_{fa} \), respectively.\(^2\) In contrast, those of malicious sensors can be arbitrarily different, denoted for the \( i \)-th malicious one by \( \gamma_1(i) P_d \) and \( \gamma_0(i) P_{fa} \), respectively. (Note that \( 0 \leq \gamma_1(i) P_d, \gamma_0(i) P_{fa} \leq 1 \) for any malicious sensor \( i \).) The factors \( \gamma_1(i) \) and \( \gamma_0(i) \) represent the difference in spectrum sensing ability between the \( i \)-th malicious sensor and honest ones. In addition, the reporting channel between each user and the fusion center is assumed to be error-free. 4) An honest sensor will send its local spectrum sensing result directly to the fusion center. Here, it is further assumed that honest secondary users are in a proximity and make independent observations of the same spectrum. Different sensing behaviors caused by sensor location variability in large networks (e.g., [77]) can be avoided/mitigated by clustering [78, 79] and are ignored in this work. 5) A malicious sensor, however, will tamper its local inference before reporting to the fusion center. In particular, the \( i \)-th malicious sensor will flip its local inference from 0 to 1 and 1 to 0 with probabilities \( \varphi_{01}(i) \) and \( \varphi_{10}(i) \), respectively, which will be referred to as flipping attack in the following discussions.

From the fusion center’s viewpoint, the equivalent detection and false alarm probabilities of the \( i \)-th malicious sensor with flipping probabilities \( \varphi(i) \triangleq [\varphi_{01}(i), \varphi_{10}(i)] \) are given by

\[
\begin{align*}
P_d^{(M,i)} &= (1 - \varphi_{10}(i))\gamma_1(i) P_d + \varphi_{01}(i)(1 - \gamma_1(i) P_d), \quad \text{(3.1)} \\
P_{fa}^{(M,i)} &= (1 - \varphi_{10}(i))\gamma_0(i) P_{fa} + \varphi_{01}(i)(1 - \gamma_0(i) P_{fa}). \quad \text{(3.2)}
\end{align*}
\]

If a malicious sensor \( i \) attacks, i.e., \( \{\varphi_{01}(i), \varphi_{10}(i)\} \neq \{0,0\} \), its statistical behavior will deviate from that of the honest sensor. The objective of this work is to detect the malicious sensors by observing their statistical deviations.

\(^2\)The superscript \( H \) is dropped for simplicity.
3.2 The Proposed Method

The proposed malicious sensor detection method consists of two parts: a conditional frequency check (CFC) and an auxiliary hamming distance check (HDC). In this section, it is assumed that the spectrum sensing capabilities of the malicious and the honest devices are identical, i.e., \( \gamma^{(i)}_1 = \gamma^{(i)}_0 = 1 \) for any malicious sensor \( i \). Extension to more general attackers with arbitrary \( \gamma^{(i)}_1 \) and \( \gamma^{(i)}_0 \) will be presented in Section 3.4.

3.2.1 Conditional Frequency Check

According to the preceding modeling, a malicious sensor has two degrees of freedom, i.e., two parameters \( \varphi_{01} \) and \( \varphi_{10} \), in launching a flipping attack. The conventional frequency check (FC), which detects malicious sensors by computing their frequencies of reporting 1, enforces only one constraint to the attacker’s behavior as indicated in Eq. (3.6) below. This is insufficient to prevent the malicious sensor from attacking. In contrast, our proposed CFC can enforce two constraints by exploring the correlation between consecutive spectrum states when the true spectrum states are Markovian, and consequently can identify any flipping attacker easily. In particular, the CFC consists of two statistics as defined below.

**Definition 1.** The two conditional frequency check statistics of a sensor are defined as \( \Psi_1 \triangleq Pr(r_t = 1|r_{t-1} = 1) \), and \( \Psi_0 \triangleq Pr(r_t = 0|r_{t-1} = 0) \), respectively, where \( r_t \) denotes the sensor’s report at time \( t \).

According to the definition, these two statistics are related to the model parameters as (see Appendix B.1)

\[
\Psi_1 = \frac{\pi_0 a_{00} P_{fa}^2 + (\pi_0 a_{01} + \pi_1 a_{10}) P_{fa} P_{fa} + \pi_1 a_{11} P_{fa}^2}{\pi_0 P_{fa} + \pi_1 P_{fa}}, \quad (3.3)
\]

\[
\Psi_0 = \frac{\pi_0 a_{00} (1 - P_{fa})^2 + (\pi_0 a_{01} + \pi_1 a_{10})(1 - P_{fa})(1 - P_{fa}) + \pi_1 a_{11}(1 - P_{fa})^2}{\pi_0 (1 - P_{fa}) + \pi_1 (1 - P_{fa})}, \quad (3.4)
\]

In the CFC, the fusion center will evaluate \( \Psi_1 \) and \( \Psi_0 \) for every sensor and compare the resulting values with those of the trusted sensor. If the values are different, the corresponding sensor will be identified as malicious. In the following, the effectiveness of this statistical check is demonstrated through two analytical results, followed by a practical approach to estimating these two statistics that eliminates the requirement of any prior knowledge about the sensing and spectrum models.

**Proposition 1.** For the Markov spectrum model considered in this work, any sensor that survives the CFC can pass the FC.
Proof. A malicious sensor can pass the FC as long as \(Pr(r_t^{(M)} = 1) = Pr(r_t^{(tr)} = 1)\), where \(r_t^{(M)}\) (\(r_t^{(tr)}\)) denotes the malicious (trusted) sensor’s report at time \(t\). However, the malicious sensor needs to satisfy both \(\Psi_0^{(M)} = \Psi_0^{(tr)}\) and \(\Psi_1^{(M)} = \Psi_1^{(tr)}\) to survive the CFC.

Note that \(Pr(r_t^{(tr)} = i) = Pr(r_t^{(tr)} = i) (i \in \{0, 1\})\) when the true spectrum states are in equilibrium, and \(Pr(r_t^{(tr)} = 1) = \Psi_1 Pr(r_{t-1}^{(tr)} = 1) + (1 - \Psi_0) Pr(r_{t-1}^{(tr)} = 0)\). Consequently, for any sensor that survives the CFC, we have

\[
Pr(r_t^{(M)} = 1) = \frac{1 - \Psi_0^{(M)}}{2 - \Psi_0^{(M)} - \Psi_1^{(M)}} = \frac{1 - \Psi_0^{(tr)}}{2 - \Psi_0^{(tr)} - \Psi_1^{(tr)}} = Pr(r_t^{(tr)} = 1),
\]

which implies that this sensor can also pass the FC. \(\square\)

**Proposition 2.** If \(\frac{a_{10} P_{fa} + a_{01} P_d}{a_{10} + a_{01}} \neq \frac{1}{2}\), a malicious sensor can never pass the CFC if it attacks, i.e., \(\{\varphi_0, \varphi_1\} \neq \{0, 0\}\). If \(\frac{a_{10} P_{fa} + a_{01} P_d}{a_{10} + a_{01}} = \frac{1}{2}\), an active malicious sensor can pass the CFC only if it sets \(\{\varphi_0, \varphi_1\} \) to \(\{1, 1\}\).

**Proof.** According to Proposition D.2, passing the FC is a necessary condition for a malicious sensor to pass the CFC. Thus, \(\varphi\) must satisfy \(\pi_0 P_{fa}^{(M)} + \pi_1 P_d^{(M)} = Pr(r_t^{(M)} = 1) = Pr(r_t^{(tr)} = 1) = \pi_0 P_{fa} + \pi_1 P_d\) to survive the CFC. Considering (3.1) and (3.2), this implies the following linear constraint on \(\varphi_0\) and \(\varphi_1\):

\[
\varphi_0 (\pi_0 (1 - P_{fa}) + \pi_1 (1 - P_d)) = \varphi_1 (\pi_0 P_{fa} + \pi_1 P_d).
\]

When (3.6) holds, define \(g_1(\varphi_10) \triangleq (\pi_0 P_{fa} + \pi_1 P_d) \cdot (\Psi_1^{(M)} - \Psi_1)\). After some algebra, it can be shown that

\[
\begin{align*}
g_1(\varphi_10) &= \varphi_0^2 \kappa_1^2 \pi_0 \pi_1 a_{00} - (\pi_1 a_{10} + \pi_0 a_{01}) \pi_1 \pi_0 + \pi_1 \pi_1^2 a_{11} + \varphi_0 \kappa_1 [-2\pi_1 \pi_0 a_{00} P_{fa} + (\pi_1 a_{10} + \pi_0 a_{01})(P_{fa} \pi_0 - P_d \pi_1) + 2\pi_1 \pi_0 a_{11} P_d],
\end{align*}
\]

where \(\kappa_1 \triangleq \frac{P_{fa} - P_d}{(\pi_0 (1 - P_{fa}) + \pi_1 (1 - P_d))}\).

Note that the malicious sensor can pass the CFC only if it could find a \(\varphi^* = [\varphi_0^*, \varphi_1^*]\) that satisfies both \(g_1(\varphi_10^*) = 0\) (i.e., \(\Psi_1^{(M)} = \Psi_1\)) and (3.6). Denote \(\varphi_1^*\) as the non-zero root of \(g_1(\varphi_10)\) = 0, which can be found as:

\[
\begin{align*}
\varphi_1^* &= -\frac{\xi_2}{\kappa_1 \xi_1},
\end{align*}
\]

where \(\xi_1 \triangleq \pi_0 \pi_1^2 a_{00} - (\pi_1 a_{10} + \pi_0 a_{01}) \pi_1 \pi_0 + \pi_1 \pi_0^2 a_{11}\) and

\[
\begin{align*}
\xi_2 &\triangleq -2\pi_1 \pi_0 a_{00} P_{fa} + (\pi_1 a_{10} + \pi_0 a_{01}) \cdot (\pi_0 P_{fa} - \pi_1 P_d) + 2\pi_1 \pi_0 a_{11} P_d.
\end{align*}
\]
According to (3.6) and (3.7), $\varphi^*_{01}$ is given by
\[ \varphi^*_{01} = -\frac{\xi_2}{\kappa_0 \xi_1}, \] (3.8)
where $\kappa_0 \triangleq \frac{P_{fa} - P_d}{(\pi_0 P_{fa} + \pi_1 P_d)}$.

Consider the relation $\pi A = \pi$, (3.7) and (3.8) can be simplified as
\[ \varphi_{10}^* = 2 - \frac{2(a_{10} P_{fa} + a_{01} P_d)}{a_{10} + a_{01}}, \] (3.9)
\[ \varphi_{01}^* = \frac{2(a_{10} P_{fa} + a_{01} P_d)}{a_{10} + a_{01}}. \] (3.10)

As a direct consequence of (3.9) and (3.10), $\varphi_{10}^* + \varphi_{01}^* = 2$ must hold if the malicious sensor wants to pass the CFC. On the other hand, $0 \leq \varphi_{01}^*, \varphi_{10}^* \leq 1$ by definition. These two conditions imply that $\{\varphi_{01}^*, \varphi_{10}^*\}$ exists only if $\frac{2(a_{10} P_{fa} + a_{01} P_d)}{a_{10} + a_{01}} = 1$ and the corresponding $\{\varphi_{01}^*, \varphi_{10}^*\}$ equals $\{1, 1\}$. Otherwise, there is no valid non-zero solution for both $g_1(\varphi_{10}) = 0$ and (3.6). That is, the malicious sensor cannot pass the CFC if it attacks.

Define the error function
\[ e(\varphi) \triangleq ||\Psi^{(tr)} - \Psi^{(M)}||_2, \] (3.11)
where $\Psi^{(tr)} \triangleq [\Psi_1^{(tr)}, \Psi_0^{(tr)}]$ and $\Psi^{(M)} \triangleq [\Psi_1^{(M)}, \Psi_0^{(M)}]$ are the CFC statistics of the trusted sensor and the malicious sensors, respectively. A typical figure of $e(\varphi)$ when the condition $\frac{a_{10} P_{fa} + a_{01} P_d}{a_{10} + a_{01}} = \frac{1}{2}$ holds is shown in Fig. 3.1. As can be seen, $\{1, 1\}$ is the only blind spot of the CFC. In contrast, the conventional FC only enforces a linear constraint (3.6) on the attacker, and thus leaves a blind line as indicated in Fig. 3.1. Furthermore, for a fixed $\varphi$, a larger $e(\varphi)$ implies a more significant statistical deviation of the malicious report as compared to the honest report and consequently easier detection. Considering this, the notion of detection margin is proposed in the following to characterize the difficulty of detecting the malicious users.

**Definition 2.** Given $P_d$ and $P_{fa}$ of honest sensors and the spectrum state transition matrix $A$, the detection margin (DM) of the CFC is defined as the maximum $e(\varphi)$ for any attacking strategy $\varphi$ that survives the frequency check. That is,
\[ DM(P_d, P_{fa}, A) \triangleq \max_{\varphi \text{ satisfies (3.6)}} e(\varphi). \] (3.12)

Remark: Seen from (3.7), for a given blind line (3.6), $(\Psi_1^{(M)} - \Psi_1)$ is a quadratic function, and
so is \((\Psi_0^{(M)} - \Psi_0)\). The DM is the peak of the corresponding quadratic curve above the blind line by definition (see Fig. 3.1). Thus for any given blind line, a larger DM implies easier detection of malicious sensors. The following proposition relates the value of the detection margin to the probabilities of detection and false alarm of honest users.

**Proposition 3.** The detection margin \(DM\) is an increasing function of \(P_d\) and \((1 - P_{fa})\) when \(P_d \geq 0.5\) and \(P_{fa} \leq 0.5\). The minimum value of \(DM\) is zero and is achieved when \(P_d = P_{fa} = 0.5\).\(^3\)

**Proof.** From (3.7), it can be seen that, for any fixed \(\phi^*\) be observed from (3.13).

From (3.7), it can be seen that, for any fixed \(P_d\), \(P_{fa}\) and \(A\), the maximum of \(e(\varphi)\) is achieved at \(\varphi_{10} = \frac{1}{2}(0 + \varphi_{10}^0)\) and \(\varphi_{01} = \frac{1}{2}(0 + \varphi_{01}^0)\) when (3.6) is satisfied, where \(\varphi_{10}^0\) and \(\varphi_{01}^0\) are given in (3.9) and (3.10) respectively. Further, it can be verified that \(\Psi_1^{(M)} = \frac{1}{2}\varphi_{01}^*\) at \(\tilde{\varphi} = \{\varphi_{10}, \varphi_{01}\}\). Consequently,

\[
\begin{align*}
|\Psi_1^{(tr)} - \Psi_1^{(M)}(\tilde{\varphi})| &= \left|\frac{\Psi_1^{(tr)} - \frac{1}{2}\varphi_{01}^*}{|\varphi_{01}^*|}ight| \\
&= \left|\frac{a_{10}a_{01}(1 - a_{01} - a_{10})(P_d - P_{fa})^2}{(a_{01} + a_{10})(a_{01}P_d + a_{10}P_{fa})}\right| \\
&= \left|\frac{a_{10}a_{01}(1 - a_{01} - a_{10})}{(a_{01} + a_{10})}\right| \cdot \frac{(P_d - P_{fa})^2}{(a_{01}P_d + a_{10}P_{fa})}.
\end{align*}
\]

(3.13)

Taking the derivative of the second term with respect to \(P_d\) leads to \(\frac{\partial}{\partial P_d} \frac{(P_d - P_{fa})^2}{(a_{01}P_d + a_{10}P_{fa})} = \frac{1}{(a_{01}P_d + a_{10}P_{fa})^2} \cdot 2a_{01}P_d + a_{10}P_{fa} \geq 0\), which implies that \(|\Psi_1^{(tr)} - \Psi_1^{(M)}(\tilde{\varphi})|\) is an increasing function of \(P_d\). It can be shown that \(|\Psi_1^{(tr)} - \Psi_1^{(M)}(\tilde{\varphi})|\) is also an increasing function of \((1 - P_{fa})\). Similarly, one can verify that \(|\Psi_0^{(tr)} - \Psi_0^{(M)}(\tilde{\varphi})|\) is an increasing function of \(P_d\) and \((1 - P_{fa})\). Therefore, \(DM = ||\Psi^{(tr)} - \Psi^{(M)}(\tilde{\varphi})||_2\) is an increasing function of \(P_d\) and \((1 - P_{fa})\). The fact that the minimum value of \(DM\) is zero and is achieved at \(P_d = P_{fa} = 0.5\) can readily be observed from (3.13). \(\square\)

Remark: Proposition 3 indicates that if honest users have better sensing quality, the detection margin is larger, which in turn implies stronger resistance to Byzantine attackers. On the other hand, a sensor network of poor quality is vulnerable to Byzantine attacks; the worst case is shown in Fig. 3.2 where \(P_d = P_{fa} = 0.5\) and \(DM = 0\). The notion of detection margin and its relation to \(P_d\) and \(P_{fa}\) will be used in Section 3.5 to explain the behaviors of CFC for different \(P_d\) and \(P_{fa}\).

\(^3\)In practice, no sensor with \(P_d \leq 0.5\) and/or \(P_{fa} \geq 0.5\) should be used.
Definition 3. For any sensor, two histogram estimators for $\Psi_1$ and $\Psi_0$ are defined as

$$
\hat{\Psi}_1 \triangleq \frac{\left( \sum_{t=1}^{T-1} \delta_{r_{t+1},1}\delta_{r_t,1} \right)}{\left( \sum_{t=1}^{T-1} \delta_{r_t,1} \right)},
$$

$$
\hat{\Psi}_0 \triangleq \frac{\left( \sum_{t=1}^{T-1} \delta_{r_{t+1},0}\delta_{r_t,0} \right)}{\left( \sum_{t=1}^{T-1} \delta_{r_t,0} \right)},
$$

respectively, where $\delta_{i,j} = 1$ iff $i = j$ and $T$ is the detection window length.

Proposition 4. The two estimators $\hat{\Psi}_1$ and $\hat{\Psi}_0$ converge to $\Psi_1$ and $\Psi_0$, respectively, as $T \to \infty$.

Proof. The proof is given in the Appendix B.2.

Remark: According to Proposition 4, the CFC statistics of all honest sensors (including the trusted one) will converge to the same value, i.e., $\Psi^{(tr)}$. On the other hand, the CFC statistics of any malicious sensor $i$ will converge to some value $\Psi^{(M,i)}$ (depending on its $\varphi^{(i)}$), which is different from $\Psi^{(tr)}$ according to Proposition D.4. Therefore, any sensor whose CFC statistic differs from that of the trusted sensor is malicious. Based on this rationale, the proposed CFC procedure is presented in Algorithm 1 where a threshold $\beta_{CFC}$ is required due to the finite detection window length in practice.

3.2.2 The Hamming Distance Check

As shown in Fig. 3.1, the CFC will fail to detect a malicious sensor employing $\varphi = \{1,1\}$ when $\frac{a_{10}P_{fa} + a_{01}P_d}{a_{10} + a_{01}} = \frac{1}{2}$. This may happen, for example, when $a_{10} = a_{01}$ and $P_d + P_{fa} = 1$. However,
$P_f a = 0.5$, $P_d a = 0.5$, $a_{01} = 0.2$, $a_{10} = 0.2$.  

\[ \phi_{01} e(\phi) = \left| \Psi_{\text{tr}} - \Psi_{i} \right|^2 \]

Figure 3.2: Typical graph of $e(\phi)$ when the condition $a_{01} P_f a + a_{10} P_d a = 1$ holds and $DM = 0$.

**Algorithm 1** The CFC procedure

1. Compute $\hat{\Psi}_{\text{tr}}^{(t)}$ and $\hat{\Psi}_{0}^{(t)}$ for the trusted sensor according to (3.14) and (3.15).
2. for sensor $i$ do
   1. Compute $\hat{\Psi}_{i}^{(t)}$ and $\hat{\Psi}_{0}^{(t)}$ according to (3.14) and (3.15).
   2. if $\left| \hat{\Psi}_{\text{tr}}^{(t)} - \hat{\Psi}_{i}^{(t)} \right|^2 > \beta_{\text{CFC}}$ then
      1. Classify sensor $i$ as malicious.
   end if
3. end for

in this case, a large normalized hamming distance, defined as

\[ d_h(i, \text{tr}) \triangleq \frac{1}{T} \sum_{t=1}^{T} \delta_{r_t^i, r_t^{(tr)}}, \]

is expected between the reported sequences from a malicious sensor $i$ and the trusted sensor, because of the high flipping probability $\phi$. Based on this observation, sensor $i$ will be identified as malicious if $d_h(i, \text{tr})$ is greater than a pre-specified threshold $\beta_{\text{HDC}}$.

### 3.3 Selection of Thresholds

This section provides guidance on the selection of the two thresholds $\beta_{\text{CFC}}$ and $\beta_{\text{HDC}}$. Specifically, since both $\hat{\Psi}$ and $d_h$ are random, $\beta_{\text{CFC}}$ and $\beta_{\text{HDC}}$ may in principle be determined by their corresponding $3\sigma$ confidence regions [80]. However, the Markovian property of the true spectrum state renders the accurate evaluation of the standard deviations difficult. In the following, some approximate results are obtained through the asymptotic independence property which has already been exploited in the proof of Proposition 4 (see the Appendix B.2).
3.3.1 Selection of $\beta_{CFC}$

Claim 1. An approximation of $\beta_{CFC}$ is given by \[ \max \left\{ 3\sqrt{\frac{\Psi_1^{(tr)}(1-\Psi_1^{(tr)})}{0.5T}}, 3\sqrt{\frac{\Psi_0^{(tr)}(1-\Psi_0^{(tr)})}{0.5T}} \right\} \]

Remark: As shown in the Appendix B.2, $\Psi_1 = \frac{1}{n_1} \sum_{i=1}^{n_1} X_{t_i}$, and $E[X_{t_i}] = \Psi_1$. Also, it can be verified that $\text{Var}[X_{t_i}] = \Psi_1(1-\Psi_1)$ due to its Bernoulli distribution. Similar observations apply to $\Psi_0$. Therefore, we can approximate the standard deviation of $\Psi_1$ and $\Psi_0$ as $\sqrt{\frac{\Psi_1-\Psi_1^2}{n_1}}$ and $\sqrt{\frac{\Psi_0-\Psi_1^2}{n_0}}$, respectively, by assuming that $X_{t_i}$’s are independent.\(^4\) Also, $\pi_0 T$ and $\pi_1 T$ can be used as estimates of $n_0$ and $n_1$ (defined in the Appendix B.2), respectively. Further, although $\hat{\Psi}_0$ and $\hat{\Psi}_1$ are correlated in general, we treat them as independent to avoid tedious computation, and use $\max\left\{ \sqrt{\frac{\Psi_1(1-\Psi_1)}{\pi_1 T}}, \sqrt{\frac{\Psi_0(1-\Psi_0)}{\pi_0 T}} \right\}$ to approximate the standard deviation of $||\hat{\Psi}||_2$, which equals the standard deviation of $||\Psi - \hat{\Psi}||_2$. In Claim 1, $\beta_{CFC}$ is chosen as the $3\sigma$ confidence interval based on this approximate standard deviation, where the true values of $\Psi$ and $\pi$ are replaced by $\Psi^{(tr)}$ and 0.5, respectively.

3.3.2 Selection of $\beta_{HDC}$

Claim 2. An approximation of $\beta_{HDC}$ is given by $P_{fa}(1-P_{fa}) + P_d(1-P_d) + 3\sqrt{\frac{f(P_{fa})+f(P_d)}{T}}$, where $f(x) \triangleq x(1-x)(1-2x+2x^2)$.

Remark: The probabilities that two honest sensors report different observations are $2P_{fa}(1-P_{fa})$ and $2P_d(1-P_d)$ when the true spectrum states are 0 and 1, respectively. Therefore, the expectation of $d_h$, the normalized hamming distance between two honest sensors, is given by $E(d_h) = 2\pi_0 P_{fa}(1-P_{fa}) + 2\pi_1 P_d(1-P_d)$. By a similar argument as in the derivation of Claim 1 above, we approximate the variance of $d_h$ by $\frac{2\pi_0 P_{fa}(1-P_{fa})(1-2P_{fa}+2P_{fa}^2) + 2\pi_1 P_d(1-P_d)(1-2P_d+2P_d^2)}{T}$. In Claim 2, $\beta_{HDC}$ is chosen as the right $3\sigma$ deviation bound of $d_h$ (i.e., $E(d_h) + 3\sigma_d$) based on this approximate variance, where the true values of $\pi_0$ and $\pi_1$ are replaced by 0.5. In practice, nominal values of $P_d$ and $P_{fa}$ can be used to compute $\beta_{HDC}$.

3.4 Extensions of the CFC

The previous analysis demonstrates the effectiveness of the proposed CFC under the assumptions that both honest and malicious users are equipped with the same spectrum sensing devices (i.e., $\gamma_0^{(i)} = \gamma_1^{(i)} = 1$ for any malicious sensor $i$) and a trusted sensor is always available. This

\(^4\)Note that $X_{t_i}$’s are actually dependent as argued in Appendix B.2. The independence assumption here is intended for simplifying the computation and obtaining an approximate threshold, which is shown to work reasonably well by simulation results in Section 3.5.
section extends the study to more general circumstances. In particular, the effectiveness of the CFC to more general attackers are shown analytically, and a complementary approach of the CFC for the case that the trusted sensor is not available is presented.

### 3.4.1 More General Attackers

In practice, malicious users may carry devices that have different spectrum sensing accuracy as compared to the honest sensors. For this reason, more general attackers are considered in this subsection. In particular, the probabilities of detection and false alarm of the \(i\)-th malicious sensors are \(\gamma_1^{(i)} P_d\) and \(\gamma_0^{(i)} P_{fa}\), respectively, where \(\gamma_1^{(i)}\) and \(\gamma_0^{(i)}\) are arbitrary so long as \(0 \leq \gamma_1^{(i)} P_d, \gamma_0^{(i)} P_{fa} \leq 1\). The following proposition shows that the proposed CFC is still effective even in the presence of such more general attackers.

**Proposition 5.** For any attacker with spectrum sensing ability \(\gamma_1 P_d\) and \(\gamma_0 P_{fa}\), the blind spot of the CFC \(\{\varphi_{01}^*, \varphi_{01}'\}\) is given by

\[
\varphi_{01}^* = \frac{2a_{01}P_d + a_{10}P_{fa}}{a_{01} + a_{10}} - \frac{\gamma_1 - \gamma_0}{\gamma_1 P_d - \gamma_0 P_{fa}}, \tag{3.17}
\]

\[
\varphi_{10}^* = 1 - \varphi_{01}^* + \frac{P_d - P_{fa}}{\gamma_1 P_d - \gamma_0 P_{fa}}. \tag{3.18}
\]

**Proof.** Define the equivalent flipping indices \(\varphi_{10}'\) and \(\varphi_{01}'\) such that

\[
P_d^{(M)} = (1 - \varphi_{10} - \varphi_{01}) \gamma_1 P_d + \varphi_{01} = (1 - \varphi_{10}' - \varphi_{01}') P_d + \varphi_{01}', \tag{3.19}
\]

\[
P_{fa}^{(M)} = (1 - \varphi_{10} - \varphi_{01}) \gamma_0 P_{fa} + \varphi_{01} = (1 - \varphi_{10}' - \varphi_{01}') P_{fa} + \varphi_{01}', \tag{3.20}
\]

where \(\varphi_{10}'\) and \(\varphi_{01}'\) are not probabilities and hence could be greater than 1. Taking the difference of (3.19) and (3.20) results in

\[
(1 - \varphi_{10}' - \varphi_{01}') = (1 - \varphi_{10} - \varphi_{01}) \frac{\gamma_1 P_d - \gamma_0 P_{fa}}{P_d - P_{fa}}. \tag{3.21}
\]

Substituting the preceding equation back into (3.20), it has

\[
\varphi_{01}' = \varphi_{01} + (1 - \varphi_{10} - \varphi_{01}) P_d P_{fa} \frac{\gamma_0 - \gamma_1}{P_d - P_{fa}}. \tag{3.22}
\]

\[
^5\text{The user index } (i) \text{ is omitted here for clarity.}
\]
By noticing the similarity between the structures of (3.1), (3.2) and (3.19), (3.20), the same approach as in the proof of Proposition D.4 can be applied here to find the blind spot of the CFC. In particular, the following two analogues of (3.9) and (3.10) can be observed

\[
\varphi'_{01} = 2 - \frac{2(a_{10}P_{fa} + a_{01}P_d)}{a_{10} + a_{01}},
\]

(3.23)

\[
\varphi'_{10} = \frac{2(a_{10}P_{fa} + a_{01}P_d)}{a_{10} + a_{01}}.
\]

(3.24)

Substituting (3.23) and (3.24) into (3.21) and (3.22) leads to (3.17) and (3.18).

Remark: In fact, Proposition D.4 is a special case of Proposition 5 when \(\gamma_0 = \gamma_1 = 1\). Further, it is worth noting that another choice for the malicious sensor to pass the CFC is to set \(\varphi_{01} = \frac{\gamma_1 - \gamma_0}{\gamma_1 \gamma_0} P_d P_{fa}\) and \(\varphi_{10} = 1 - \varphi_{01} + \frac{P_d - P_{fa}}{\gamma_1 \gamma_0 P_d - \gamma_1 \gamma_0 P_{fa}}\), which corresponds to a flipping indices pair \(\{\varphi_{01}, \varphi_{10}\} = \{0, 0\}\). But it can be verified that \(P_d^{(M)} = P_d\) and \(P_{fa}^{(M)} = P_{fa}\) in this case. That is, the malicious sensor behaves statistically the same as the honest sensor.

**Corollary 1.** For a malicious sensor with weaker spectrum sensing ability than the honest ones, the CFC has no blind spot. (Specifically, \(\gamma_1\) and \(\gamma_0\) of a weaker malicious sensor admit \(\gamma_1 < 1\), \(\gamma_0 \geq 1\), or \(\gamma_1 = 1\), \(\gamma_0 > 1\).)

**Proof.** In this case, it can be verified that surviving the CFC requires \(\varphi_{01}^* > 2\) and \(\varphi_{10}^* > 2\) according to (3.17) and (3.18), which implies invalid \(\varphi_{01}^\ast\) and \(\varphi_{10}^\ast\).

Remark: For a malicious sensor more powerful than the honest ones, the existence of the blind spot of the CFC depends on the specific values of corresponding sensing ability and true spectrum parameters.

**Proposition 6.** The expectation of the normalized hamming distance between a CFC-surviving malicious sensor \(m\) and the trusted sensor \(E[d_h(m, tr)]\) is no less than that between a honest sensor \(h\) and the trusted sensor \(E[d_h(h, tr)]\). The expectation of the distance gap \(E[d_h(m, tr)] - E[d_h(h, tr)]\) is irrespective of \(\gamma_1^{(m)}\) and \(\gamma_0^{(m)}\), and is lower bounded by \((P_d - P_{fa}) \cdot \min(2P_d - 1, 1 - 2P_{fa})\), which is non-trivial when \(P_d > 0.5\) and \(P_{fa} < 0.5\).

**Proof.** Note that given any two sensors \(i\) and \(j\) with equivalent spectrum sensing capabilities \(P_d^{(i)}, P_{fa}^{(i)}\) and \(P_d^{(j)}, P_{fa}^{(j)}\), the expected normalized hamming distance is given by

\[
E[d_h(i, j)] = \sum_{k=0}^{1} \pi_k \left[ Pr(r_{t}^{(i)} = 1, r_{t}^{(j)} = 0|s_t = k) + Pr(r_{t}^{(i)} = 0, r_{t}^{(j)} = 1|s_t = k) \right]
\]

\[
= \pi_0 \left[ P_{fa}^{(i)} (1 - 2P_{fa}^{(j)}) + P_{fa}^{(j)} \right] + \pi_1 \left[ P_d^{(i)} (1 - 2P_d^{(j)}) + P_d^{(j)} \right].
\]

(3.25)
Thus,

$$E[d_h(m, tr)] - E[d_h(h, tr)] = \pi_0\left(P_{fa}^{(M, m)}(1 - 2P_{fa}) + P_{fa}\right) + \pi_1\left(P_d^{(M, m)}(1 - 2P_d) + P_d\right)$$

$$- \pi_0\left(P_{fa}(1 - 2P_{fa}) + P_{fa}\right) - \pi_1\left(P_d(1 - 2P_d) + P_d\right).$$

Since the malicious sensor survives the CFC, one can substitute (3.19), (3.20), (3.23) and (3.24) into the preceding equation, which leads to

$$E[d_h(tr, m)] - E[d_h(tr, h)] = 2(P_d - P_{fa})\frac{a^2_{10}(2P_d - 1) + a^2_{01}(1 - 2P_{fa})}{(a_{10} + a_{01})^2}$$

$$\geq (P_d - P_{fa})\cdot \min(2P_d - 1, 1 - 2P_{fa}).$$

(3.27)

Remark: Although the CFC may have a blind spot to non-weaker malicious sensors and fail to detect them, Proposition 6 implies that the HDC can be employed to detect such malicious sensors effectively as long as $(P_d - P_{fa})\cdot \min(2P_d - 1, 1 - 2P_{fa}) > 0$, which is generally true in practice.

### 3.4.2 In the Absence of Trusted Sensor

When there is no trusted user in the network and the fusion center is distant from the sensing spot (and thus cannot serve as a sensor), a heuristic clustering method shown in Algorithm 2 can be used to detect the malicious sensors when honest users dominate the network (accounting for more than 50% of the total number of nodes). The intuitive idea behind Algorithm 2 is that two users $i$ and $j$ will fall into the same cluster if their corresponding CFC statistics satisfy $|\Psi^j - \Psi^i| \leq \beta_{CFC}$. The condition $|\Psi^j - \Psi^i| \leq \beta_{CFC} (\forall j \in C)$ ensures that any two users within the same cluster will not deviate from each other dramatically. As a result, $\Psi$’s of honest sensors will form a cluster with approximate radius $\beta_{CFC}/2$. In addition, due to the honest user domination assumption, the cluster formed by the honest sensors always has larger cardinality than any cluster formed by malicious sensors.

Since Algorithm 2 may fail to filter out some malicious sensors due to the existence of the blind spot, a modified HDC is presented in Algorithm 3 to catch these malicious sensors. Particularly, the modified HDC divides the sensors that survive Algorithm 2 into two groups $U_1$ and $U_2$ by

\[\text{Since no trusted sensor is available, } \beta_{CFC} \text{ is computed by assuming that } \Psi_{0}^{tr} = \Psi_{1}^{tr} = 0.5.\]
Algorithm 2 User classification by clustering

Start with the set of all sensors $U = \{1, 2, ..., N\}$.

while $U \neq \emptyset$ do
    Pick a random $k \in U$.
    Create a new cluster $C = \{k\}$.
    $U = U \setminus \{k\}$.
    for $i \in U$ do
        if $\exists j \in C$ such that $|\Psi(j) - \Psi(i)| \leq \frac{\beta_{CFC}}{2}$ and $\forall j \in C$, $|\Psi(j) - \Psi(i)| \leq \beta_{CFC}$ then
            $C = C \cup \{i\}$, $U = U \setminus \{i\}$.
        end if
    end for
end while

Sensors in the cluster of the largest cardinality are honest.

the corresponding normalized hamming distances. By the assumption that the honest sensors dominate the network, the one of $U_1$ and $U_2$ with larger cardinality will be identified as the honest sensor set. In particular, in Algorithm 3, $H$ and $M$ denote the sets of honest sensors and malicious sensors identified by Algorithm 2, respectively. In addition, $|H|$ denotes the cardinality of the set $H$. Note that $|M| > |H|$ may happen, which implies that although the effectiveness of Algorithm 3 is ensured only when $|M| < |H|$, it may work for certain cases with $|M| > |H|$. Specifically, such case happens when the CFC statistics of malicious sensors form multiple clusters with each having a cardinality less than $|H|$.

Algorithm 3 Modified HDC

Start with $H = \{h_1, h_2, \ldots\}$.
Pick a random $k$ ($1 \leq k \leq |H|$).
Set $U_1 = H$ and $U_2 = \emptyset$.
for $i = 1 : |H|$ do
    if $d_k(h_k, h_i) \geq \beta_{HDC}$ then
        Move $h_i$ from $U_1$ to $U_2$.
    end if
end for
if $|U_1| \geq |U_2|$ then
    Malicious sensors: $M = U_2 \cup M$.
    Honest sensors: $H = U_1$.
else
    Malicious sensors: $M = U_1 \cup M$.
    Honest sensors: $H = U_2$.
end if
3.5 Simulations

Throughout the simulations, each malicious sensor randomly selects its own \( \{\varphi_{01}, \varphi_{10}\} \) uniformly over \((0, 1]^2\). The total number of honest and malicious sensors are denoted by \(n_H\) and \(n_M\), respectively. In addition, it is assumed that the first honest sensor is the trusted one, which is known by the fusion center. The thresholds \(\beta_{CFC}\) and \(\beta_{HDC}\) are selected according to Claim 1 and Claim 2, respectively. At the fusion center, AND rule, OR rule and majority voting are all valid options for data fusion. Among them, the AND rule tends to exploit the primary channel aggressively with the risk of introducing excessive inference to the primary users while OR rule is the opposite; majority voting lies in between. As the proposed malicious user detection method is invariant to the specific fusion rule, any one of these data fusion rules can be used. Specifically, majority voting is adopted in the following simulations. Fig. 3.3 depicts the block diagram of the proposed algorithm. As shown, the most recent \(T\) spectrum sensing history of all users \(r_t\)'s are stored for malicious user detection, and the fusion center will determine the current spectrum state estimate \(\hat{s}_t\) based on current sensing reports \(r_t\)'s and the detected user types. (The probabilities of detection and false alarm at the fusion center will be denoted as \(P_{d}\) and \(P_{fa}\), respectively, in the following.)

3.5.1 Basic Examples

Two different cases are simulated in this subsection to demonstrate how the proposed method effectively detects the malicious sensors and enhances the spectrum sensing results accordingly. In both cases, it is assumed that all malicious users are equipped with the same sensing devices as honest ones (i.e., \(\gamma_{0}^{(i)} = \gamma_{1}^{(i)} = 1\) for any malicious sensor \(i\)), and specifically \(P_d = 0.9\) and \(P_{fa} = 0.1\). In the first case, \(A = [0.8, 0.2, 0.2, 0.8]\), while in the second case, \(A = [0.8, 0.2, 0.4, 0.6]\). Thus, the condition \(a_{10}P_{fa} + a_{01}P_d = \frac{1}{2}\) is satisfied in the first case but not in the second one. There are \(n_H = 8\) honest sensors and \(n_M = 13\) malicious sensors, i.e., the malicious sensors dominate the network. The detection window length is \(T = 100\) (time slot).

Simulation results of a typical run of the first case are shown in Fig. 3.4–Fig. 3.6. By comparing Fig. 3.4 and Fig. 3.5, it can be seen that two malicious sensors whose flipping probabilities \(\varphi_{01}\) and \(\varphi_{10}\) are close to 1 successfully pass the CFC. However, these two malicious
Figure 3.4: Malicious sensor detection result using CFC in the first case.

Table 3.1: Average performance comparison over 1000 runs.

<table>
<thead>
<tr>
<th></th>
<th>No detection</th>
<th>Trusted only</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_d^*$ (case one)</td>
<td>0.921</td>
<td>0.900</td>
<td>0.995</td>
</tr>
<tr>
<td>$P_{fa}^*$ (case one)</td>
<td>0.075</td>
<td>0.102</td>
<td>0.001</td>
</tr>
<tr>
<td>$\eta$ (case one)</td>
<td>(___)</td>
<td>(___)</td>
<td>96.9%</td>
</tr>
<tr>
<td>$P_d^*$ (case two)</td>
<td>0.928</td>
<td>0.900</td>
<td>0.996</td>
</tr>
<tr>
<td>$P_{fa}^*$ (case two)</td>
<td>0.077</td>
<td>0.099</td>
<td>0.001</td>
</tr>
<tr>
<td>$\eta$ (case two)</td>
<td>(___)</td>
<td>(___)</td>
<td>96.7%</td>
</tr>
</tbody>
</table>

sensors fail to pass the succeeding HDC. Also, it can be seen by comparing Fig. 3.5 and Fig. 3.6 that there is one malicious user surviving both CFC and HDC. Further examination reveals that the flipping probabilities of this malicious user are low: $\phi_{01} \approx 0$ and $\phi_{10} \approx 0.1$. Although this malicious sensor is not detected, its negative influence on the spectrum sensing result of the fusion center is negligible.

Fig. 3.7–Fig. 3.8 show results of a typical run of the second case where $a_{10}P_{fa} + a_{01}P_d \neq \frac{1}{2}$. As expected, the CFC alone successfully detects the severe attacker without activating the HDC, because the CFC has no blind point in this case and can detect any active attacker according to Proposition D.4.

Table 3.1 summarizes the simulation results over 1000 Monte Carlo runs for these two cases. As can be seen, in both cases, the proposed algorithm (using both CFC and HDC) provides high malicious sensor detection accuracy ($\eta > 95\%$) with a detection window of length 100 time slots. Accordingly, it also achieves nearly perfect spectrum sensing results in both cases, i.e., $P_d = 0.995$ and $P_{fa} = 0.001$ in the first case, and $P_d = 0.996$ and $P_{fa} = 0.001$ in the second case, which are significantly better than the performance of using only the trusted sensor and that of using all sensors without malicious user detection.
Figure 3.5: Malicious sensor detection result using CFC and HDC in the first case.

Figure 3.6: True sensor types in the first case.

Figure 3.7: Malicious sensor detection result using CFC in the second case.
3.5.2 Further Simulations

This subsection will demonstrate how user classification accuracy and spectrum sensing performance of the proposed method are affected by 1) the detection window length $T$, 2) the spectrum sensing ability of the honest sensors, i.e., $P_d$ and $P_{fa}$, and 3) the percentage of malicious sensors in the network $\rho$. For this purpose, three scenarios are considered, where $n_H = 8$, $n_M = 13$, $P_d = 0.8$ and $P_{fa} = 0.2$ in the first scenario, $n_H = 8$, $n_M = 13$, $P_d = 0.95$ and $P_{fa} = 0.05$ in the second scenario, and $n_H = 8$, $n_M = 73$, $P_d = 0.8$ and $P_{fa} = 0.2$ in the last one. The spectrum state transition matrix $A = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}$ in all these three scenarios. The corresponding simulation results are shown in Fig. 3.9–Fig. 3.12, respectively, where each point in the figure accounts for 1000 Monte Carlo runs.

It is worth noting that the optimal spectrum sensing performances by assuming that the
As can be seen from Fig. 3.9–Fig. 3.12, increasing the detection window length $T$ improves both the spectrum sensing performance and user classification accuracy of the fusion center. When $T$ is sufficiently large, the classification accuracy is high ($\geq 95\%$), and thus the corresponding spectrum sensing performances are very close to the theoretical optimal ones given in (3.28) and (3.29).

Comparing Fig. 3.9 and Fig. 3.10, it can be seen that the required detection window length $T$ for similar performance is much smaller in the scenario 2, as compared to scenario 1. The same observation can be made in Fig. 3.12. In fact, it can be verified that the corresponding blind lines are the same in these two scenarios while the detection margin in scenario 2 is higher as a consequence of larger $P_d$ and smaller $P_{fa}$ (according to Proposition 3). That is, for the same attacking strategy $\varphi$, the statistical deviations $e(\varphi)$ of malicious sensors will be more significant.

\footnote{In this work, only hard-decision malicious user detection is considered. That is, a secondary user is identified either as an honest one or as a malicious one, and the reports from malicious users will be discarded. The analysis for the soft-decision case remains a future work.}
in scenario 2, and thus a shorter detection window will suffice.

Naturally the adversary may attempt to enhance the attack through increasing the number of malicious sensors. Considering this, the percentage of malicious sensors is increased from $\rho = 62\%$ in scenario 1 to $\rho = 90\%$ in scenario 3. As can be seen from Fig. 3.11 and Fig. 3.12, the proposed algorithm can still provide appealing classification accuracy and spectrum sensing performances by slightly increasing the detection window length, even though the adversary dramatically increases the amount of malicious sensors. Consequently, the adversary’s attempt to launch more severe flipping attack by increasing the number of malicious sensors is ineffective for our scheme.
Table 3.2: Average performance comparison over 1000 runs for more general malicious sensors.

<table>
<thead>
<tr>
<th></th>
<th>No detection</th>
<th>Trusted only</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P^*_d$ (strong attacker)</td>
<td>0.916</td>
<td>0.901</td>
<td>0.996</td>
</tr>
<tr>
<td>$P^*_fa$ (strong attacker)</td>
<td>0.080</td>
<td>0.099</td>
<td>0.001</td>
</tr>
<tr>
<td>$\eta$ (strong attacker)</td>
<td>(\bot)</td>
<td>(\bot)</td>
<td>95.0%</td>
</tr>
<tr>
<td>$P^*_d$ (weak attacker)</td>
<td>0.936</td>
<td>0.900</td>
<td>0.994</td>
</tr>
<tr>
<td>$P^*_fa$ (weak attacker)</td>
<td>0.074</td>
<td>0.099</td>
<td>0.001</td>
</tr>
<tr>
<td>$\eta$ (weak attacker)</td>
<td>(\bot)</td>
<td>(\bot)</td>
<td>98.3%</td>
</tr>
</tbody>
</table>

3.5.3 Malicious Users with More/Less Powerful Sensors

In practice, malicious users may be equipped with different spectrum sensing devices hoping to further disrupt the fusion center. For this reason, this subsection explores the performance of the proposed method when more general malicious sensors exist. In particular, assume that $\gamma_1 = 1.1$, $\gamma_0 = 0$ for strong attackers and $\gamma_1 = 0.8$, $\gamma_0 = 1.1$ for weak attackers. Other parameters remains the same as the first case in Section 3.5.1. The corresponding results are summarized in Table 3.2. It can be seen that the proposed algorithm still can detect malicious sensors with high accuracy and provide satisfactory sensing performance, which justifies the analytical results in Section 3.4.1.

3.5.4 Removing the Trusted Sensor

When the trusted sensor is not available, the fusion center can adopt Algorithm 2 and Algorithm 3 to detect the malicious sensors so as to ensure the robustness of collaborative spectrum sensing. Table 3.3 summarizes the results over 1000 runs for such a circumstance where $P_d = 0.8$ and $P_{fa} = 0.2$, $\gamma_0^{(i)} = \gamma_1^{(i)} = 1$ for all malicious sensor $i$, and $A = \begin{bmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{bmatrix}$. There are 5 severe attackers with $\varphi_{01} = \varphi_{10} = 1$ while other attackers select their $\varphi$’s uniformly over $(0, 1)^2$. As can be seen, even when $\rho = 45\%$ sensors in the network (of 21 sensors) are malicious, the proposed algorithm still can detect them satisfactorily ($\eta = 92.5\%$) and improve the spectrum sensing performance from $P^*_d = 0.864$, $P^*_fa = 0.142$ to $P^*_d = 0.975$, $P^*_fa = 0.012$.

Table 3.4 summarizes the results of an even worse case, where the trusted sensor is not available and the malicious users are equipped with more advanced sensing devices ($\gamma_0^{(i)} = 0$ and $\gamma_1^{(i)} = 1.2$ for all malicious sensor $i$). However, as can be seen from Table 3.4, the proposed algorithm can still detect the malicious user accurately and improve the spectrum sensing performance significantly. The similar performances shown in Table 3.3 and Table 3.4 justify the robustness of the proposed algorithm to more powerful malicious sensors even when the trusted sensor is not available.
Table 3.3: Performance of the proposed method without the trusted sensor over 1000 runs.

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>25%</th>
<th>30%</th>
<th>35%</th>
<th>40%</th>
<th>45%</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\eta$</td>
<td>94.1%</td>
<td>93.7%</td>
<td>93.6%</td>
<td>92.9%</td>
<td>92.5%</td>
</tr>
<tr>
<td>$P_d^*$ (proposed)</td>
<td>0.992</td>
<td>0.990</td>
<td>0.987</td>
<td>0.983</td>
<td>0.975</td>
</tr>
<tr>
<td>$P_d^*$ (no detection)</td>
<td>0.960</td>
<td>0.944</td>
<td>0.924</td>
<td>0.893</td>
<td>0.864</td>
</tr>
<tr>
<td>$P_{fa}^*$ (proposed)</td>
<td>0.003</td>
<td>0.005</td>
<td>0.006</td>
<td>0.009</td>
<td>0.012</td>
</tr>
<tr>
<td>$P_{fa}^*$ (no detection)</td>
<td>0.040</td>
<td>0.055</td>
<td>0.076</td>
<td>0.104</td>
<td>0.142</td>
</tr>
</tbody>
</table>

Table 3.4: Performance of the proposed method without the trusted sensor over 1000 runs when malicious sensors are more powerful.

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>25%</th>
<th>30%</th>
<th>35%</th>
<th>40%</th>
<th>45%</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\eta$</td>
<td>94.9%</td>
<td>94.3%</td>
<td>93.8%</td>
<td>92.8%</td>
<td>91.5%</td>
</tr>
<tr>
<td>$P_d^*$ (proposed)</td>
<td>0.993</td>
<td>0.993</td>
<td>0.988</td>
<td>0.984</td>
<td>0.974</td>
</tr>
<tr>
<td>$P_d^*$ (no detection)</td>
<td>0.928</td>
<td>0.901</td>
<td>0.864</td>
<td>0.832</td>
<td>0.784</td>
</tr>
<tr>
<td>$P_{fa}^*$ (proposed)</td>
<td>0.004</td>
<td>0.005</td>
<td>0.005</td>
<td>0.008</td>
<td>0.011</td>
</tr>
<tr>
<td>$P_{fa}^*$ (no detection)</td>
<td>0.084</td>
<td>0.115</td>
<td>0.146</td>
<td>0.182</td>
<td>0.247</td>
</tr>
</tbody>
</table>

3.6 Conclusions

With Markovian modeling of spectrum states, a new Byzantine attack detection method, which consists of two natural but effective CFC statistics and an auxiliary HDC, is proposed in this chapter. In addition, two consistent histogram estimators of these two statistics are developed so that no prior information on sensing and spectrum models is required in the proposed method. Both theoretical analysis and simulation results show that the proposed method is effective even when the malicious users are equipped with more advanced sensing devices. With the assistance of one trusted sensor, the proposed method can provide high detection accuracy and achieve near optimal collaborative spectrum sensing performance for arbitrary percentage of malicious sensors. In the case when the trusted sensor is not available, the extended algorithms can still maintain satisfactory detection and enhance the sensing performance significantly when the honest sensors dominate the network.
Chapter 4

Dynamic Adaptive Anti-Jamming via Controlled Mobility

Besides the static attacks such as the Byzantine attack discussed in previous chapters, dynamic and adaptive attacks, in which the attacker can astutely change their attacking strategies in accordance with the defense mechanisms, also appear due to the advancement in both software and hardware. This evolution in attacking strategy changes the conventional static security problems to more sophisticated dynamic security competitions. To illustrate this, the dynamic jamming/anti-jamming competition between a mobile relay network and a group of mobile jammers considered in this chapter serves as a good example, where the relay nodes continuously reconfigure their geographic locations to improve the network performance, and at the same time, the intelligent jammers also respond by moving to more effective jamming positions. Particularly, the problem of how to control mobility for single and multiple commodity flow optimizations will be considered. For these two problems, new spectral quantities, single- and multi-weighted Cheeger constants and corresponding eigenvalue variants, are constructed based on spectral graph theory [24], which can nicely relate network performance metrics to the spectral properties of the network graph and provide a promising design basis for mobility and topology control with global performance assurance. Both analytical results and simulations are presented to demonstrate that relays and jammers can compete effectively by directing their motions according to these spectral quantities of the network graph. In addition, application in mobile CR networks is presented as well.

The rest of this chapter is organized as follows. Section 4.1 describes the general problem setting and notations. The single and the multiple commodity flow problems under jamming are treated in Section 4.2 and 4.3, respectively. Numerical results are presented in Section 4.4. Section 4.5 discusses related works. Conclusions are given in Section 4.6.
4.1 Problem Formulation and Notations

In wireless networks consisting of mobile nodes, one interesting application is that multiple mobile relays help the source(s) forward data to distant destination(s) in the presence of intelligent mobile jammers who intend to disrupt the S-D communications. This work focuses on the case where the sources and destinations are static and considers both single-flow and multiple-flow problems. Jammers are intentionally endowed with more power to have full position information of all legitimate nodes and thus can intelligently respond to the relay motions. It is also assumed that the relays can employ jammer localization techniques (e.g., [81–83]) to continuously estimate jammers’ positions and adapt to it;\(^1\) the influence of localization error will be discussed in Section 4.4.1.

Notations used throughout this chapter are in order: 1) \(n\) is the number of legitimate nodes (including the source(s), destination(s) and relays) that form the vertices of the network graph and, \(l\) is the number of jammers. 2) The positions of the legitimate nodes and the jammers are denoted by \(\{x_i = (x_i(1), x_i(2)) \in \mathbb{R}^2 : i = 1, ..., n\}\) and \(\{x_J = (x_J(1), x_J(2)) \in \mathbb{R}^2 : J = 1, ..., l\}\), respectively. 3) The Laplacian matrix of the corresponding network graph \(L \triangleq D - A\) is determined by the positions of legitimate nodes and the jammers. In particular, \(A \triangleq [a_{i,j}]_{i,j=1}^n\) is the generalized adjacency matrix with \(a_{i,j}\) given in (4.1) below, and \(D \triangleq diag\{\delta_1, ..., \delta_n\}\) is the generalized degree matrix with \(\delta_i \triangleq \sum_{j \neq i} a_{i,j}\). 4) The edge capacity \(a_{i,j}\) is taken as the 1-nat (i.e., \(\log_2 e\) bits) information exchanging time between nodes \(i\) and \(j\), i.e.,

\[
a_{i,j} \triangleq \begin{cases} 
B \cdot \left( \frac{1}{\ln(1 + SIR_{i,j})} + \frac{1}{\ln(1 + SIR_{j,i})} \right)^{-1}, & (i \neq j), \\
0, & (i = j),
\end{cases}
\]

(4.1)

where \(B\) is the system bandwidth, and \(SIR_{i,j}\) is the signal-to-interference ratio at node \(j\) for the transmission from \(i\). In our study, \(SIR_{i,j}\) is defined as

---

\(^1\)Continuously monitoring the jammers may incur extra power and computation cost. The study of the optimal tradeoff between the performance gain and the cost remains a future work.
\[ SIR_{i,j} \triangleq \frac{d_{i,j}^{-\alpha}}{\eta \sum_{J=1}^{J} d_{j,J}^{-\alpha} + \sum_{k \in \Xi_{i,j}} u(d_{j,k}/r_{int})}, \tag{4.2} \]

where the first term in the denominator stands for the interference from the jammers, and the second term encapsulates the constraint on the interactions among mobile relays, further illustrated below, and \( SIR_{j,i} \) is similarly defined.

In (4.2), the path loss model (with exponent \( \alpha \)) is used for signal propagation as in [2, 84], and the effects of random channel attenuations due to fading will be discussed in Section 4.4. Besides, \( d_{i,j} = ||x_i - x_j||_2 \) is the distance between nodes \( i \) and \( j \), and \( d_{i,J} = ||x_i - x_J||_2 \) and \( d_{j,J} = ||x_j - x_J||_2 \) are their distances to jammer \( J \), respectively, while \( \eta \) is used to denote the jammer-to-legitimate node power ratio. The term \( \sum_{k \in \Xi_{i,j}} u(d_{j,k}/r_{int}) \), where \( \Xi_{i,j} \) denotes the set of all legitimate nodes excluding \( i \) and \( j \), is included to prevent relays from getting too close to each other, which either violates physical constraints or introduces excessive self-interference. In particular, the following smoothed step function (as depicted in Fig. 4.1) is considered:

\[ u(y) \triangleq \xi \cdot \frac{\exp(-\kappa y - \log y_0)}{1 + \exp(-\kappa y - \log y_0)}, \tag{4.3} \]

with \( y_0 \) an arbitrarily small positive number (e.g., \( y_0 = 10^{-3} \)) and \( \xi \) and \( \kappa \) design parameters. The parameter \( r_{int} \) sets the interference zone around each relay.

This model captures several important factors in wireless networks. In practice, it can be modified to accommodate channel and application specifics, without changing the essence of our design presented below.

### 4.2 Single Flow under Jamming

This section investigates anti-jamming mobility control in the context of the single commodity flow problem, starting from a natural naive approach based on direct application of existing spectral quantities, the original Cheeger constant and the second smallest eigenvalue of the network graph Laplacian matrix. After revealing the inefficiency in this naive approach, new spectral quantities, single-weighted Cheeger constant and the corresponding eigenvalue variant, are constructed. Analytical results imply that these quantities bring about more effective anti-jamming performance and certain level of global performance assurance.

In the information flow context [85], each edge \((i,j)\) of the network graph receives a flow
that cannot exceed its capacity \( a_{i,j} \). In addition, for each node, the incoming and outgoing flows should be equal, except for the sources and destinations. In the single commodity flow problem, it is assumed that there is only one S-D pair and the objective is to find the maximum achievable flow \( F_{\text{sgl}} \) from the source to the destination. Due to the well-known max-flow min-cut theorem [86], \( F_{\text{sgl}} \) always equals to the single flow min-cut \( C_{\text{sgl}} \) of the underlying network graph. In particular, \( C_{\text{sgl}} \) measures the network flow bottleneck and is define as:

\[
C_{\text{sgl}} = \min_{\{S: v_s \in S, v_d \notin S\}} \sum_{i \in S, j \notin S} a_{i,j}, \tag{4.4}
\]

where \( v_s \) is the source and \( v_d \) is the destination node, and \( S \subset \{1, \ldots, n\} \) represents a subset of nodes. The specific goal of the relays (jammers) is to increase (decrease) the maximum achievable S-D flow \( F_{\text{sgl}} \) through properly controlled movements.

### 4.2.1 Preliminary on the Cheeger Constant

The Cheeger constant (a.k.a. network conductance [87]) is a measure of the bottleneck of a network graph. In spectral graph theory, it is usually defined for the normalized Laplacian matrix \( \mathcal{L} \triangleq D^{-1/2}LD^{-1/2} \) as [24]

\[
h(\mathcal{L}) = \min_S \frac{\sum_{i \in S, j \notin S} a_{i,j}}{\min\{\text{vol}(S), \text{vol}(\bar{S})\}}, \tag{4.5}
\]

where \( \text{vol}(S) = \sum_{i \in S} \delta_i \). Associated with this quantity is the normalized Cheeger’s inequality

\[
\lambda_2(\mathcal{L})/2 \leq h(\mathcal{L}) \leq \sqrt{2\lambda_2(\mathcal{L})}. \tag{4.6}
\]

This analytical result nicely relates the global performance metric \( h(\mathcal{L}) \) with the second smallest eigenvalue \( \lambda_2(\mathcal{L}) \) of the network Laplacian matrix, which can be optimized by controlled node mobility as will be discussed in Section 4.2.4.

### 4.2.2 A Naive Approach

The combinatorial optimization form in (4.4) makes it very challenging to obtain a direct analytical relation between the node positions and the min-cut \( C_{\text{sgl}} \), such that the corresponding mobility control can be easily implemented. The Cheeger constant and associated inequality provide us some clues: \( h(\mathcal{L}) \) and \( C_{\text{sgl}} \) are similar in terms of both mathematical definition
and physical meaning, and $\lambda_2(\mathcal{L})$ can serve as a useful approximation for $h(\mathcal{L})$. However, it is worth noting that $h(\mathcal{L})$ has a scaling issue for the flow problems we consider: it is invariant to the scaling of the link capacities $\{a_{i,j}\}$ while our targets $C_{sgl}$ and $F_{sgl}$ apparently are not. To address this issue, the Cheeger constant of the unnormalized Laplacian matrix $\mathbf{L}$ can be employed, which is defined as [88]

$$h(\mathbf{L}) = \min_S \frac{\sum_{i \in S, j \notin S} a_{i,j}}{\min\{|S|, |\bar{S}|\}}, \quad (4.7)$$

where $|S|$ is the cardinality of $S$; it admits the unnormalized Cheeger’s inequality

$$\lambda_2(\mathcal{L})/2 \leq h(\mathbf{L}) \leq \sqrt{2\delta_{\max}}\lambda_2(\mathcal{L}), \quad (4.8)$$

where $\delta_{\max}$ is the maximum vertex degree. Although $h(\mathcal{L})$ and $h(\mathbf{L})$ coincide when $a_{i,j}$’s are restricted to $\{0, 1\}$, $h(\mathbf{L})$ is free from the scaling issue and thus is more suitable here.

Given above considerations, a natural but naive approach for both the legitimate nodes and the jammers is to direct their motions based on $\lambda_2(\mathcal{L})$, hoping to change $F_{sgl}$ correspondingly. In [89], $\lambda_2(\mathcal{L})$ was exploited for network connectivity maintenance in the presence of jamming. However, the problem of this approach is that $h(\mathbf{L})$ fails to fully capture the real intentions of the attacker and the defender in the single commodity flow problem, and so is the corresponding $\lambda_2(\mathcal{L})$. Specifically, $h(\mathbf{L})$ measures the bottleneck of the entire network but not necessarily the bottleneck of the specific S-D flow. Consequently, using this approach, relays will blindly aim at improving the weakest links in the network and may fail to protect the desired S-D flow; similarly, it may also mislead the jammers and cause less effective attack.

The above discussion indicates that direct application of existing spectral quantities may not be suitable for our goal. We thus advocate the concept of application-oriented spectral quantity design as shown in the following subsection.

### 4.2.3 Single-weighted Cheeger Constant

In this subsection, a single-weighted Cheeger constant $h_W(\mathbf{L})$ and a corresponding eigenvalue variant $\lambda_2(\mathbf{L}_W)$ are constructed specifically for the single commodity flow problem, and an associated Cheeger-like inequality is also derived as a theoretical support for performance assurance.

In the single commodity flow problem, both legitimate relays and the jammers are particularly interested in the maximum data flow between the specific source and destination, and care less about the bottleneck between other pair of nodes. However, unlike the min-cut defined in
(4.4), the \( h(L) \) defined in (4.7) does not enforce the optimal cut \((S, \bar{S})\) separating the source \( v_s \) and the destination \( v_d \), and thus fails to capture the intentions of the relays and the jammers. This can be addressed if the definition of \( h(L) \) is modified as:

\[
h(L) = \min_{\{S: v_s \in S, v_d \in \bar{S}\}} \frac{\sum_{i \in S,j \in \bar{S}} a_{i,j}}{\min\{|S|, |S|\}}.
\]

(4.9)

But this change of definition destroys the favorable connection between \( h(L) \) and \( \lambda_2(L) \) promised by the Cheeger’s inequality. To resolve this issue, a single-weighted Cheeger constant \( h_W(L) \) is constructed as follows:

\[
h_W(L) = \min_S \frac{\sum_{i \in S,j \in \bar{S}} a_{i,j}}{\min\{|S|_W, |S|_W\}},
\]

(4.10)

where the weighted cardinality \(|S|_W \triangleq \sum_{i \in S} w_i\) and \( w_i \geq 0 \) is the weight assigned to node \( i \). It can be readily noticed that (4.10) approximate (4.9) with a suitable weight assignment: \( w_u = 1 \) for \( u \in \{v_s, v_d\} \) and \( w_u = 0 \) for \( u \not\in \{v_s, v_d\} \), which also conforms to intuition as only the source and destination have data exchanging demands. But this weighting results in non-invertible weight assignment matrix \( W \triangleq diag\{w_1, ..., w_n\} \), causing numerical issues when the mobility control method in Section 4.2.4 is applied. In practice, a proper way for weight assignment is to associate the source and destination nodes with larger \( w \)’s and relays with smaller \( w \)’s while keeping \( W \) invertible (e.g., the weight assignment used in Section 4.4).

Similar to the previous naive approach, it is desirable to have a suitable (variant of) network eigenvalue to bridge \( h_W(L) \) and the corresponding mobility control. To this end, a variant of \( \lambda_2(L) \) is proposed, which is defined as the second smallest eigenvalue of the weighted matrix \( L_W \triangleq W^{-1/2}LW^{-1/2} \):

\[
\lambda_2(L_W) \triangleq \inf_{g \perp W^{1/2}} \frac{g^T L_W g}{g^T g}.
\]

(4.11)

Furthermore, the following Cheeger-like inequality holds.

**Proposition 7.** Define \( w_{\min} \triangleq \min_i w_i \), then \( \lambda_2(L_W) \) and \( h_W(L) \) admit

\[
\frac{\lambda_2(L_W)}{2} \leq h_W(L) \leq \sqrt{2\delta_{\max} \lambda_2(L_W) / w_{\min}}.
\]

(4.12)

**Proof.** Please see Appendix C.1. \( \square \)
The above proposition indicates that the relays can relocate themselves to maximize $\lambda_2(LW)$ such that the lower bound of $h(L)$ increases; while the jammers can maneuver to minimize $\lambda_2(LW)$ such that the upper bound of $h(L)$ decreases. Since, as discussed above, $h(L)$ is more appropriate than $h(L)$; these facts imply that $\lambda_2(LW)$ may be a more appropriate mobility control metric for the single commodity flow optimization, as compared to $\lambda_2(L)$, which will be further justified by numerical results in Section 4.4.

4.2.4 Mobility Control Based on $\lambda_2(LW)$

To maximize $\lambda_2(LW)$, a relay $i$ can move in the direction of the spatial gradient of $\lambda_2(LW)$ with respect to its current position $x_i$, i.e., moving in the direction $(\partial \lambda_2(LW) / \partial x_i^{(1)}, \partial \lambda_2(LW) / \partial x_i^{(2)})$. Specifically, the spatial gradient can be computed as

$$\frac{\partial \lambda_2(LW)}{\partial x_i^{(k)}} = \left\langle \frac{\partial \lambda_2(LW)}{\partial LW}, \frac{\partial LW}{\partial x_i^{(k)}} \right\rangle = y^T \frac{\partial LW}{\partial x_i^{(k)}} y$$

(4.13)

where the inner product of two matrices is defined by $\langle A, B \rangle \triangleq \text{trace}(A^T B)$ and the fact that $\frac{\partial \lambda_2(LW)}{\partial LW} = yy^T$ is applied with $y$ the eigenvector corresponding to $\lambda_2(LW)$ and $y_p (y_q)$ the $p$th ($q$th) component of $y$; $p \sim q$ means that nodes $p$ and $q$ are connected. The detailed expression of $\frac{\partial a_{p,q}}{\partial x_i^{(k)}}$ is given in Appendix C.2. In contrast, jammers aim at minimizing $\lambda_2(LW)$ and thus will move in the direction opposite to their spatial gradients, i.e., $\left( -\frac{\partial \lambda_2(LW)}{\partial x_j^{(1)}}, -\frac{\partial \lambda_2(LW)}{\partial x_j^{(2)}} \right)$ for $J \in \{1, ..., l\}$. It is also worth noting that the expression in the last step of (4.13) admits a distributed implementation by employing the consensus-based distributed eigenvalue/eigenvector estimation technique in [90].

4.3 Multiple Flows under Jamming

In this section, we extend our study to the multiflow scenario. A brief review of the multi-commodity flow problem, particularly the maximum concurrent flow [91], is given first, and the proposed multi-weighted Cheeger constant follows, along with the corresponding eigenvalue variant for mobility control. In addition, the corresponding Cheeger inequality is derived to
support the expected jamming/anti-jamming performance analytically.

### 4.3.1 The Multicommodity Flow Problem

The maximum concurrent flow is defined as the maximum value of \( f_{\text{mult}} \) such that \( f_{\text{mult}} D^{(m)} \) units of commodity \( m \) (1 \( \leq m \leq M \)) can be simultaneously routed without violating any edge capacity constraints, where \( D^{(m)} \) denotes the demand for the \( m \)th commodity [91]. The optimal objective value of the following linear programming gives the maximum concurrent flow \( F_{\text{mult}} \):

\[
\begin{align*}
\max_{x(+)\!, \!x(-)} & \quad f_{\text{mult}} \\
\text{subject to} & \quad \sum_{j=1}^{n} \left( x(+)_{i,j}^{(m)} - x(-)_{i,j}^{(m)} \right) = f_{\text{mult}} D^{(m)}_i, \forall i, m, \\
& \quad \sum_{m=1}^{M} \left( x(+)_{i,j}^{(m)} + x(-)_{i,j}^{(m)} \right) \leq a_{i,j}, \forall i, j, \\
& \quad x(+)_{i,j}^{(m)}, x(-)_{i,j}^{(m)} \geq 0, \forall i, j, m,
\end{align*}
\]

where \( D^{(m)}_i \) is the demand for the \( m \)th commodity at node \( i \) (which equals \( -D^{(m)} \) \( D^{(m)} \)) if node \( i \) is the source (destination) for commodity \( m \) and zero otherwise; \( x(+)_{i,j}^{(m)} \) and \( x(-)_{i,j}^{(m)} \) denote the flows of commodity \( m \) coming from \( j \) to \( i \) and going out from \( i \) to \( j \), respectively.

The corresponding minimum multicut \( C_{\text{mult}} \) is defined as [91]:

\[
C_{\text{mult}} = \min_S \frac{\sum_{i \in S, j \in S} a_{i,j}}{\sum_{m \in \mathcal{M}(S)} D^{(m)}},
\]

where \( \mathcal{M}(S) \triangleq \{ m : v_{s_m} \in S \text{ and } v_{d_m} \in S \} \), or \( v_{s_m} \in S \text{ and } v_{d_m} \in S \}, \) and \( v_{s_m} \) is the source while \( v_{d_m} \) is the destination node for commodity \( m \).

The relays and jammers will compete for \( F_{\text{mult}} \) by changing \( \{ a_{i,j} \} \) through spatial relocations. Based on a similar rationale as in the single commodity case, the minimum multicut \( C_{\text{mult}} \) will be used as a replacement of \( F_{\text{mult}} \) for mobility regulation, though the relation between \( F_{\text{mult}} \) and \( C_{\text{mult}} \) is known only up to scaling: \( \Omega(\frac{C_{\text{mult}}}{\log n}) \leq F_{\text{mult}} \leq C_{\text{mult}} \) [91], and finding more precise bounds remains an open problem.

### 4.3.2 Multi-weighted Cheeger Constant

Now we need to explore how the relays and the jammers can control their motions to contend for \( C_{\text{mult}} \). To this end, again, it is desirable to construct a suitable variant of network eigenvalue that can be used to bound \( C_{\text{mult}} \) (if determining a more precise relation is difficult), so as to
bridge the optimization of $C_{\text{mult}}$ and the corresponding mobility control. This is a non-trivial problem. To the best of our knowledge, the most recent progress in this aspect is the theoretical work in [92], where a variant of the minimum multicut is constructed and related bounds are derived. Although these bounds have theoretical merits, they are not directly related to (variants of) network eigenvalues, which makes them unsuitable for practical mobility control. Besides the $C_{\text{mult}}$ defined in (4.15), it is also intuitively appealing to consider the following variant (with $1_{\mathcal{M}(S)}(m) = 1$ if $m \in \mathcal{M}(S)$ and zero otherwise)

\[
C'_{\text{mult}} = \min_S \sum_{i \in S, j \in \bar{S}} a_{i,j} \min_{1 \leq m \leq M} 1_{\mathcal{M}(S)}(m) \cdot D(m),
\]

(4.16)

which enforces that the corresponding optimal cut $(S, \bar{S})$ separates all S-D pairs.

To investigate the more suitable one, or a proper combination, of $C_{\text{mult}}$ and $C'_{\text{mult}}$, a multi-weighted Cheeger constant $h_{W}^{(M)}(L)$ with parameter $k \in [0, 1)$ that can capture the features of both of them is constructed in this work. Particularly, associate each of the $M$ flows with a weight assignment matrix $W^{(m)} = \text{diag}\{w_{1}^{(m)}, ..., w_{n}^{(m)}\}$ and define

\[
h_{W}^{(M)}(L) = \min_S \sum_{i \in S, j \in \bar{S}} a_{i,j} \sum_{m=1}^{M} \beta_{m} \min \{|S|_{W^{(m)}}, |\bar{S}|_{W^{(m)}\!}\},
\]

(4.17)

where $\beta_{m} \triangleq \min^{-k}\{|S|_{W^{(m)}}, |\bar{S}|_{W^{(m)}\!}\}$ for $m = 1, ..., M$. It can be verified that when $k = 0$, $h_{W}^{(M)}(L)$ reduces to $C_{\text{mult}}$ by setting $w_{i}^{(m)} = D^{(m)}$ if node $i$ is the source or the destination of flow $m$ and zero otherwise. While, for $k > 0$, $h_{W}^{(M)}(L)$ tends to capture the characteristics of $C'_{\text{mult}}$, with the following intuition: To approximate $C'_{\text{mult}}$, ideally, we want the denominator of $h_{W}^{(M)}(L)$ to have the form $\min_{1 \leq m \leq M} \min \{|S|_{W^{(m)}}, |\bar{S}|_{W^{(m)}\!}\}$. However, the minimization over $m$ causes difficulties for later eigenvalue variant construction. To address this, it is first noticed that this desired form can be expressed as a linear combination $\sum_{m=1}^{M} \beta_{m} \min \{|S|_{W^{(m)}}, |\bar{S}|_{W^{(m)}\!}\}$ where the minimum bottleneck takes coefficient 1 while other terms take coefficients 0. To avoid the difficulty of identifying the minimum bottleneck among all flows, it is further noticed that $\min^{-k}\{|S|_{W^{(m)}}, |\bar{S}|_{W^{(m)}\!}\}$ is inversely proportional to the $i$th flow bottleneck, which well approximates the desired weighting. Note that when $k$ approaches 1, the weighting effects of $W^{(m)}$’s unfavorably diminish, as can be seen from (4.17). Numerical experiments will be conducted in Section 4.4 to explore the most suitable $k$.

Based on $h_{W}^{(M)}(L)$, an eigenvalue variant $\lambda_{2}(h_{W}^{(M)})$ (with parameter $k$) is constructed as follows:
\[
\lambda_2(L^{(M)}_{\tilde{W}}) \triangleq \inf_{g \perp L^{1/2}} g^T L^{(M)}_{\tilde{W}} g = \inf_{f \perp \tilde{W}_{1/2}} f^T L f, \tag{4.18}
\]

where \(\tilde{W} \triangleq \text{diag}\{\tilde{w}_1, ..., \tilde{w}_n\}\) with \(\tilde{w}_i \triangleq \sum_{m=1}^{M} (w^{(m)}_i)^{1-k}\) and \(L^{(M)}_{\tilde{W}} \triangleq \tilde{W}^{-1/2} L \tilde{W}^{-1/2}\). Furthermore, the following Cheeger-like inequality holds.

**Proposition 8.** Define \(\tilde{w}_{\text{min}} \triangleq \min_i \tilde{w}_i\) and \(\xi^{(M)}_{\tilde{W}} \triangleq \frac{\max_m (\sum_{i=1}^{n} (w^{(m)}_i)^{1-k})}{\min_{i,m} (w^{(m)}_i)^{1-k}}\), for some constant \(k\) between 0 and 1. \(\lambda_2(L^{(M)}_{\tilde{W}})\) and \(h^{(M)}_{\tilde{W}}(L)\) admit

\[
\lambda_2(L^{(M)}_{\tilde{W}})/2 \leq h^{(M)}_{\tilde{W}}(L) \leq \sqrt{2\delta_{\text{max}} n^2 k (\xi^{(M)}_{\tilde{W}})^2 \lambda_2(L^{(M)}_{\tilde{W}})/\tilde{w}_{\text{min}}}. \tag{4.19}
\]

**Proof.** Please see Appendix C.3. \(\square\)

Based on the above results, in the multicommodity flow problem, the relays and the jammers can employ \(\lambda_2(L^{(M)}_{\tilde{W}})\) as the metric to direct their motions, and better performances are expected, as compared to using \(\lambda_2(L)\). This will be further justified by simulation results in Section 4.4.

### 4.4 Numerical Results and Discussion

In this section, numerical experiments are conducted to demonstrate 1) the significance of constructing a suitable spectral quantity for effective mobility control, and 2) the adaptive competition between the relays and the jammer(s) through controlled spatial reconfigurations. In Section 4.4.3, the proposed spectral quantities are further employed to direct secondary relay movements in CR networks. Both single and multiple commodity flow problems are investigated. To avoid numerical issues in computing \(W^{-1}\), for each commodity \(m\), \(w^{(m)} = n/2\) is assigned to the source and destination and \(w^{(m)} = 1\) to other nodes, in the proposed spectral quantities. To obtain smooth trajectories, each timeslot is divided into 10 sub-timeslots, and all relays adjust their moving directions at the end of each sub-timeslot. Other parameters used throughout the simulations are as follows: two design parameters in (4.3) \(\xi = 1\) and \(\kappa = 10\), jammer (PU)-to-relay power ratio \(\eta = 10\), the maximum speed of both relays and jammers \(\nu_{\text{max}} = 0.5\), bandwidth \(B = 1\text{Hz}\), \(r_{\text{int}} = 0.5\). All average flows are obtained through 100 Monte Carlo runs. In all figures, nodes move in the direction toward the markers.
4.4.1 Single Commodity Flow

For single commodity flow, assume that there are \( n = 10 \) legitimate nodes (including the source and the destination) and one jammer. First, the jammer is assumed static, and then a responsive mobile jammer, which intelligently changes its position to degrade the achievable S-D flow, will be considered. For given positions of the relays and the jammer, the achievable S-D flow can be computed according to the Ford-Fulkerson algorithm [86].

**Static jammer**

![Figure 4.2: Single flow and static jammer.](image)

Initially, relays are uniformly located on the line between the source and the destination as shown in Fig. 4.2a. Due to the jammer nearby, the achievable S-D flow \( F_{sgl} \) is only 0.4116 in the beginning. Then, relays start to relocate themselves to increase the achievable S-D flow as shown in Fig. 4.2b. Eventually, relays move to desirable locations corresponding to the current jammer position such that the achievable S-D flow is raised to \( F_{sgl} = 0.7279 \), as shown in Fig. 4.2c. The corresponding evolvement of \( F_{sgl} \)'s over time is shown in Fig. 4.2d when node motions are...
directed by the original eigenvalue $\lambda_2(L)$ and the proposed single-weighted eigenvalue variant $\lambda_2(LW)$, respectively. As shown, the performance of using $\lambda_2(LW)$ significantly outperforms that of using $\lambda_2(L)$. In addition, it is worth noting that $F_{sgl}$ even degrades in the initial relay adaptations when $\lambda_2(L)$ is used. The reason is that, in this case, relays blindly target protecting the weakest links in the whole network, which may not be helpful to or even sacrifice the desirable S-D flow. In contrast, when $\lambda_2(LW)$ is adopted, relays always prioritize the designated S-D flow optimization, leading to monotonic increase in $F_{sgl}$.

Mobile jammer

A responsive mobile jammer is considered first which only starts to move after the relays have settled down and started to communicate, to reduce the risk of being detected. To model such a reactive behavior, it is assumed that, in every $\tau (=10)$ sub-timeslots, the jammer first changes its direction based on the positions of the relays and moves for one sub-timeslot, and then keeps static for the rest $\tau - 1$ sub-timeslots.

To prevent the jammer from disrupting the S-D flow by simply moving arbitrarily close to either the source or the destination (assumed fixed in this study), guard zones are set up for
Table 4.1: Comparison of flows and power efficiencies.

<table>
<thead>
<tr>
<th>Method</th>
<th>Avg. Flow</th>
<th>Power Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using $\lambda_2(\mathbf{L})$, NPS</td>
<td>0.1420</td>
<td>0.0284</td>
</tr>
<tr>
<td>Using $\lambda_2(\mathbf{L}_W)$, NPS</td>
<td>0.3473</td>
<td>0.0695</td>
</tr>
<tr>
<td>Using $\lambda_2(\mathbf{L}_W)$, PS</td>
<td>0.3120</td>
<td>0.2432</td>
</tr>
</tbody>
</table>

both the source and the destination. When the jammer moves inside the guard zone, it has a risk of being caught which is assumed to be inversely proportional to its distance to the source or the destination. The jammer decides its moving direction based on both the spatial gradient discussed in Section 4.2.4 and the risk of being caught in the guard zone. Specifically, when the jammer is inside the guard zone of the source, denoting the direction vector from the source to the jammer as $\mathbf{e}_{S,J}$, the corresponding distance as $d_S = ||\mathbf{x}_S - \mathbf{x}_J||_2$, and the jammer’s spatial gradient vector computed by (4.13) as $\mathbf{g}_J$, the jammer will move in the direction $\frac{\gamma}{d_S}\mathbf{e}_{S,J} + (-\mathbf{g}_J)$ where the risk factor $\gamma$ is a design parameter and is set to 1 in the following simulations.

Fig. 4.3 demonstrates the dynamic adaptive competition process between the relays and the jammer. At the beginning, relays are located on a line as shown in Fig. 4.3a. Then the jammer moves toward the source $S_1$ to decrease the achievable S-D flow as shown in Fig. 4.3b. In response to the jammer’s action, relays relocate to advantageous positions as shown in Fig. 4.3b and 4.3c, guided by the proposed weighted conductance and the original unweighted conductance, respectively. As can be seen, while relays in Fig. 4.3c mainly aim at evading the jammer, those in Fig. 4.3b also endeavor to remain close to the source and destination; this explains the difference in the achievable S-D flow: 0.5359 vs 0.2338. As the relays adaptively retreat, the jammer pursues, which eventually forces the relays to change their moving directions (Fig. 4.3d) till reaching new advantageous positions (Fig. 4.3c). This dynamic adaptive game continues over time. The evolvement of the achievable S-D flow $F_{sgl}$’s presented in Fig. 4.3f evidences the adaptive competition between the relays and the jammer; as well as the significant performance improvement brought by the proposed weighted conductance. The power saving (PS) variant of our proposed scheme shown in Fig. 4.3f will be discussed below.

Comparing with existing method

To further demonstrate the advantages of the proposed method, it is compared with the method in [2]. As shown in Fig. 4.2d and Fig. 4.3f, the proposed method substantially outperforms that in [2].

The main reason is that the method in [2] is not originally designed for the S-D flow

---

2When simulating the curves “Ref. [2]” in Fig. 4.2d and Fig. 4.3f, we deliberately disturbed the initial relay positions along the $y$-direction slightly. The reason is that the method in [2] only allows nodes move toward other nodes, and hence with the exact initial positions in Fig. 4.2a, relays cannot move along the $y$-direction. This would have resulted in even worse performance, which we consider as an unfair comparison.
optimization, and hence, as shown in Fig. 4.4, relays only heuristically attempt to optimize their local communications (e.g., blindly moving away from the jammer), which does not necessarily improve, or even degrades, the S-D flow.

![Figure 4.4: Single flow using the method in [2].](image)

Before stepping into the multiflow case, we further explore several interesting issues in our proposed mobility control scheme; the insights obtained here readily carry over to the next section.

**Improve power efficiency:** In previous simulations, both relays and the jammer always move with full speed $\nu_{\text{max}}$. Such a moving strategy, coined as non-power saving (NPS) in Fig. 4.3f, intends to maximize the achievable data flow but may also result in low power efficiency, especially when the gradients (given by (4.13)) are small. To balance flow optimization and power consumption, a power saving variant is further considered. In particular, each relay records the maximum gradient $||g_{\text{max}}||_2$ that it has been experienced up to the current time $t$, and sets the current moving speed as $\frac{\nu_{\text{max}}||g_t||_2}{||g_{\text{max}}||_2}$, where $g_t$ is the current gradient. It can be seen from Fig. 4.3f that the S-D flow evolvement using this PS strategy outperforms the unweighted case significantly and uniformly. Taking the traversed distance of each node as (proportional to) its consumed power and achievable S-D flow-to-consumed power ratio as the power efficiency, the average achievable S-D flows and power efficiencies of different strategies are summarized and compared in Table 4.1. As can be seen, the PS strategy dramatically increases the power efficiency (from 0.0284 and 0.0695 to 0.2432) and only sacrifices the average achievable S-D flow slightly (from 0.3473 to 0.3120, which is still significantly higher than the flow in the unweighted case 0.1420).

**Localization accuracy:** Since the relays have to continuously track the jammer’s position so as to adapt appropriately, the influence of jammer localization errors deserves investigation.
For this purpose, S-D flows achieved by relay relocations under different jammer localization errors are compared in Fig. 4.5. As can be seen, the achievable flows remain almost the same for small relative localization errors (i.e., absolute localization error normalized by the average node-to-jammer distance) and degrade gracefully when the error increases. While on the other hand, a jammer tracking algorithm that can achieve localization accuracy \( \Phi_{loc} \equiv |1 - \frac{d_{eJ}}{d_{sJ}}| \times 100\% \) with \( d_{eJ} \) and \( d_{sJ} \) the estimated and true sensor-to-jammer distances, respectively) about 99.3% is reported in [81]. Together with the simulation results in Fig. 4.5, this implies that in practice relays can employ existing algorithms for jammer tracking without much performance degradation.

**Different jammer moving frequency \( \tau \):** Besides the responsive jammer with moving frequency \( \tau = 10 \), it is also interesting to consider jammers with other moving frequencies. In Fig. 4.6, the corresponding achievable S-D flows are compared for different \( \tau \)'s, where, in particular, \( \tau = 1 \) represents an aggressive jammer that continuously changes its position and moves. As
can be seen, a small $\tau$ usually induces a sharp flow degradation (in average) due to more agile adaptations of the jammer. In addition, different $\tau$’s cause different dynamisms in the S-D flows, which correspond to different relay-jammer competition processes. For instance, when the weighted conductance is used, $\tau = 20$ (i.e., moving for one sub-timeslot in every two timeslots) causes dynamisms of the S-D flows while $\tau = 1$ leads to an equilibrium. But in all cases, using the proposed eigenvalue variant $\lambda_2(LW)$ to direct relay movements always leads to significant S-D flow improvements as compared to using $\lambda_2(L)$.

![Figure 4.7: Performance comparison for Rayleigh fading channels.](image)

**Random channel attenuation:** In realistic environments, besides the path loss effect considered in (4.2), random channel attenuation due to multipath propagation and scattering also exists, which is often modelled by Rayleigh fading. Therefore the true SIR at node $j$ when receiving from node $i$ should be modified as

$$\text{STR}_{i,j} \triangleq \frac{R_{i,j}^{-\alpha} d_{i,j}^{-\alpha \gamma}}{\eta \sum_{J=1}^{t} R_{j,J} d_{j,J}^{-\alpha \gamma} + \sum_{k \in \Xi_{i,j}} u(d_{j,k}/r_{\text{int}})},$$

(4.20)

where $R_{i,j}$’s and $R_{j,J}$’s are independent Rayleigh random variables with unit mean. Correspondingly, the true link capacity $\bar{a}_{i,j}$ should be modified by replacing SIR with $\text{STR}$ in (4.1). To obtain optimal mobility control, relays must have perfect predictions of the fading coefficients $R_{p_1,p_2}(t+1)$ between any two spatial points $p_1$ and $p_2$ at the next time slot $t+1$, which is usually infeasible in practice. Note that perfect instantaneous measurements of the fading coefficients at current positions $R_{i,j}(t)$’s, even if available, are not useful for mobility control.

Further simulations have been conducted to examine the performance of the proposed scheme in the presence of random channel attenuation. The histograms of 1000 Monte Carlo
runs with both static and mobile jammers and Rayleigh fading are presented in Fig. 4.7, where relay mobility control still assumes the original path loss model. For each Monte Carlo run, the time-average maximum flow \( F_{sgl} \) over the simulated time duration \( T \) is considered as the metric of interest. As compared to the case without fading, the mean performance of the proposed method degrades a bit under fading. Nonetheless, it still provides significant performance gain over that of using \( \lambda_2(L) \), in both static (Fig. 4.7a) and mobile (Fig. 4.7b) jammer cases.

Mobile source or destination: It should be noted that the proposed method directly applies when the source and/or destination are also mobile. To avoid that the source and destination trivially move towards each other, the mobility of the destination is restricted to the \( y \)-axis and the source is static in this simulation. By applying the proposed method, the destination node intelligently moves away from the jammer and the relays relocate themselves accordingly, as shown in Fig. 4.8, in both static and mobile jammer cases. As expected, this additional mobility of the destination provides further performance gains as compared to the static destination case, as shown in Fig. 4.2d and Fig. 4.3f. Similar gains can be observed when the source is also mobile; the corresponding results are omitted in the interest of space.

4.4.2 Multicommodity Flow

Two typical multicommodity flow communication settings, multi-unicast and multicast, are investigated in this subsection with \( n = 20 \). A static jammer is assumed first and then the multiple mobile jammers case follows. The maximum achievable concurrent flow \( F_{mult} \) can be computed by solving (4.14).
Multi-unicast

First, a 2-unicast with initial deployment given in Fig. 4.9a is considered and the achievable maximum concurrent flows $F_{\text{mult}}$’s of using $\lambda_2(L)$ and $\lambda_2(L^{(M)})$ are compared in Fig. 4.10. As can be seen, through controlled mobility relays effectively adapt to the current jammer position and dramatically increase the concurrent flow $F_{\text{mult}}$ from 0.06 to 0.64 (with $k = 0$). In addition, using $\lambda_2(L^{(M)})$ leads to more effective relay relocations against the jammer, and the best performance is obtained when $k = 0$. After adaptation, relays eventually stop at the positions shown in Fig. 4.11, and it can be observed from Fig. 4.11a that, when $\lambda_2(L^{(M)})$ is used, more relays are favorably gathered around $S_1$ and $D_1$ to compensate for the SIR loss caused by the jammer. Meanwhile, relays also stay closer to $S_2$ and $D_2$ so as to help the communication in between, as compared to the unweighted case (Fig. 4.11b).

To explore further, multiple mobile jammers are considered for the 2-unicast case. Two mobile jammers and relays are initially located as shown in Fig. 4.12a, and then relays start...
to retreat and jammers begin to pursuit as shown in Fig. 4.12b. Comparing Fig. 4.12c and Fig. 4.12d, it can be seen that using the proposed conductance more relays properly distributed around $S_1$ and $D_2$ (which are under severe attacks by the two jammers nearby), while in contrast, relays largely focus on avoiding the jammers and fail to protect these two critical nodes in the unweighted case (Fig. 4.12d).

The $F_{\text{mult}}$ evolutions of using $\lambda_2(L^{(M)}_{\tilde{W}})$ and $\lambda_2(L)$ in the mobile jammer case are compared in Fig. 4.13, which again demonstrates the effectiveness of using the weighted conductance. In addition, it is shown that the single-weighted eigenvalue variants $\lambda_2(L^{(M)}_{W(i)})$ ($i = 1, 2$) are not suitable for and may even degrade the multicommodity concurrent flow. In these cases, relays only aim at protecting a single S-D flow and the other flow is subject to severe jamming attacks.

Multicast

We then consider a 1-to-3 multicast with initial deployment given in Fig. 4.9b. By comparing Fig. 4.14a and Fig. 4.14b, it can be observed that part of the relays will move closer to the source $S$ to compete against the jammer and others are distributed more appropriately to facilitate the multicommodity flow when $\lambda_2(L^{(M)}_{\tilde{W}})$ is used; while in contrast relays simply focus on avoiding the jammers, links among them will become the network bottleneck, and fail to protect the desired S-D flows when $\lambda_2(L)$ is used. In addition, $\lambda_2(L^{(M)}_{W})$ with $k = 0$ leads to the most effective mobility control as compared with other choices of $k$ and the unweighted case, as shown in Fig. 4.15. Similar observations to Fig. 4.14 can be made from Fig. 4.16, where relays are competing with two mobile jammers, and Fig. 4.17 shows the corresponding evolvement of $F_{\text{mult}}$’s and again evidences the effectiveness of using $\lambda_2(L^{(M)}_{W})$. 
4.4.3 Application in Cognitive Radio Networks

Another important application of the proposed mobility control scheme is in cognitive radio networks. In this case, a set of mobile relays is deployed in secondary networks to facilitate designated secondary flows, now facing the threat from mobile PUs instead. As compared to the anti-jamming application, two important differences are worth mentioning. On the one hand, instead of intelligently attacking the S-D flows, primary users are treated as unintended jamming sources; in this study they are assumed to move according to the well-known random waypoint model [93] (with speed $\nu_{PU}$’s uniformly chosen from $[0, 1]$). On the other hand, the communication of primary users needs adequate protection; in this study a virtual guard zone (with radius $r = 2$) is set around each primary user, and all secondary nodes (including sources, destinations and relays) that are inside any of the virtual guard zones must stop ongoing communications immediately. Both single-flow and multiple-flow cases are explored in this setting.

In the single flow case shown in Fig. 4.18, when the secondary relays detect the SIR loss of $D_1$ caused by the approaching PU, they move closer to $D_1$ assisting the desired communication,
as shown in Fig. 4.18a. When destination node $D_1$ is in the primary user’s guard zone, it stops the communication, resulting in zero secondary flow, as shown in Fig. 4.18b. The average secondary S-D flow $F_{sgl}$’s under the guidance of the proposed and the original spectral quantities are compared in Fig. 4.19. As can be seen, using the proposed weighted conductance doubles the average secondary S-D flows.

For the 2-unicast case shown in Fig. 4.20, two mobile primary users are considered, moving independently in the field. As compared to using $\lambda_2(L)$ (shown in Fig. 4.20b), the proposed weighted eigenvalue $\lambda_2(\hat{L}_W^{(M)})$ strikes a good balance between jamming resistance and communication promotion, resulting in a significant flow increment (from 0.1320 to 0.2635). Similarly, in the 1-to-3 multicast case shown in Fig. 4.22, it can be observed by comparing Fig. 4.22a and Fig. 4.22b that the proposed weighted eigenvalue $\lambda_2(\hat{L}_W^{(M)})$ direct the relays to more favorable positions. Furthermore, the corresponding average flow performance $F_{mult}$ is presented in Fig. 4.21 and Fig. 4.23, respectively, where significant improvements in $F_{mult}$ by using $\lambda_2(\hat{L}_W^{(M)})$
4.5 Related Works

In literature, various types of anti-jamming strategies have been proposed, ranging from stochastic matching (e.g., [94]), coding (e.g., [95]), directive antenna and beamforming (e.g., [96]), to spread spectrum (SS) techniques including frequency hopping (FH) and direct-sequence (DS) [26, 97]. More recent advances in this area include the uncoordinated spread spectrum techniques such as UFH [27] and UDSSS [98] for insider jamming defense, the USD-FH scheme [28] for jamming-resilient key establishment, the RD-DSSS scheme [99] and the DSD-DSSS scheme [100] for efficient keyless anti-jamming broadcast communications, and the techniques of creating covert and low data rate channels to defend strong and reactive jammers [101, 102]. In wireless networks, multinode diversity has been exploited in [103] to develop CUFH, a collaborative anti-jamming technique, and spatial diversity has been exploited in [104] to design jamming-resilient geographic routing. Recently, node mobility adds another new ingredient to the jamming/anti-jamming game. Spatial retreat was considered in [105, 106] for jamming avoidance. Based on some inspiring heuristics and a simulation study, node mobility is investigated in [2] for both the adversarial network and the legitimate network. As compared to [2], our work is based on more solid theoretical foundations and thus admits performance assurance.

4.6 Conclusions

Anti-jamming mobility control has been explored for the single and multicommodity flow problems in this work. To address the inefficiency of direct application of some existing spectral
quantity, two network graph eigenvalue variants that can characterize the target performance more appropriately are constructed. In addition, Cheeger-like inequalities are derived which show that the resulting controlled mobility has certain level of jamming resilience assurance. Simulation results are provided to demonstrate the dynamic adaptive competition between the relays and the jammer, and confirm the effectiveness of the proposed approach. In addition, the application of the proposed approach in CR networks is also demonstrated.
Figure 4.17: Comparison of $F_{\text{mult}}$’s for 1-to-3 multicast (mobile jammers).

Figure 4.18: Secondary relay trajectories for single flow.

Figure 4.19: Comparison of average single secondary flow $F_{\text{sgl}}$’s.
Figure 4.20: Comparison of secondary relay trajectories for 2-unicast.

Figure 4.21: Comparison of average secondary maximum concurrent flow $F_{\text{mult}}$’s for 2-unicast.

Figure 4.22: Comparison of secondary relay trajectories for 1-to-3 multicast.
Figure 4.23: Comparison of average secondary maximum concurrent flow $F_{\text{mult}}$'s for 1-to-3 multicast.
Chapter 5

Faster Learning and Adaptation in Security Games by Exploiting Information Asymmetry

In the dynamic jamming/anti-jamming competition considered in the previous chapter, both the legitimate network and the adversary adjust their next moves based on the current actions taken by the opponent. It is interesting to make one step further to consider security games where the defender and the attacker can not only adapt to the opponent’s current strategy but also think ahead and conduct foresighted planning. For such problems, game theory, which provides a formal analytical framework with a set of mathematical tools to study the complex interactions among rational players, seems a natural tool. Recently, there has been significant growth in research activities that use game theory for analyzing network security problems (e.g., [29, 30] and references therein). However, classical game theory and direct application of it usually assume a static environment, while the arms race between the defender and attacker in practice often takes place in (unknown) dynamic environments. Variations in wireless environments and available communication and computational resources are possible sources of such dynamics. This suggests that, in practical applications, legitimate systems (LS) have to not only meticulously deal with the intelligent adversary but also carefully accommodate their strategies in accordance to these environmental dynamics on-the-fly, so as to ensure normal system operation. With this consideration, a stochastic game (SG) [33] formulation of the dynamic security competition is considered in this chapter. Further considering that, as discussed in Chapter 1, in practical security games, each player usually has only incomplete information about the opponent, which induces information asymmetry, two new MARL algorithms, termed minimax-PDS and WoLF-PDS, are proposed, which enable the defense system to learn and adapt faster in unknown dynamic environments by exploiting its information advantage.
The contributions of this work are as follows: 1) To the best of our knowledge, the proposed algorithms are the first to explore such information asymmetry for faster learning in SG settings. 2) The proposed learning algorithms are general and admit various applications. Anti-jamming problems in EHCS and CR systems, and resource scheduling in cloud-based security games are demonstrated as concrete examples. 3) The proposed minimax-PDS and WoLF-PDS are provably convergent and rational, respectively.

The remainder of this chapter is organized as follows. Section 5.1 first presents an abstract security game model under the SG framework and then introduces the basics of conventional MARL algorithms. The proposed algorithms and their applications in EHCS, CR and cloud-based security systems are presented in Section 5.2 and Section 5.3, respectively. Simulation results are presented in Section 5.4, and related works are discussed in Section 5.5. Section 5.6 concludes this chapter.

5.1 Problem Formulation and Background

5.1.1 Problem Formulation

In the context of SG [33], an abstract security game between an LS and an attacker (opponent) can be characterized by the tuple \(< S, A, O, T, R, R^O >\), where \(S\) stands for the state space (e.g., the channel states); \(A\) and \(R\) denote the action space (e.g., the transmit powers and channels to select) and the reward function (e.g., the throughput) of the LS, respectively, while \(O\) and \(R^O\) denote those of the attacker; and \(T: S \times A \times O \rightarrow p(S)\) is the state transition function that maps the current state \(s \in S\) and the joint actions of the LS and attacker \((a, o) \in A \times O\) into a distribution \(p(s'|s, a, o)\) of the future state \(s' \in S\). As shown in Fig. 6.1, the LS adopts MARL to learn a policy \(\pi: S \rightarrow p(A)\) such that a long-term performance objective is maximized in the dynamic environment. Usually the long-term objective is given in the form of an expected cumulative discounted reward \(\mathbb{E}\left(\sum_{n=0}^{\infty} \beta^n R_n\right)\) with discounting factor \(\beta \in [0, 1)\), and similar descriptions also apply to the attacker. At each time step \(n\), both the LS and the attacker observe

\(^{1}\)We abuse the notation a bit by using \(R\) and \(R^O\) for both reward functions and random rewards.
the current state $s_n \in S$ and take actions $a_n \in A$ and $o_n \in O$, respectively, according to their own learned policies $\pi_n$ and $\pi^O_n$. Then the LS receives an immediate reward $r_n$ determined by $s_n$ and the action pair $(a_n, o_n)$. To capture the attacker’s intention of degrading the LS’s reward and without loss of generality, the common zero-sum assumption is adopted throughout this work, which assumes that the attacker’s reward is the opposite of the LS’s reward (i.e., $R^O = -R$ and $r^O_n = -r_n$). Meanwhile, the environment transits to a new state $s_{n+1} \in S$ according to the (often unknown) state transition function $T$.

5.1.2 Conventional MARL

Minimax-Q [107] is a well-known MARL algorithm for policy learning in SG. In this algorithm, the optimal quality function $Q^*_n(s, a, o)$ of a state-action pair $(s, a, o)$ for the LS is defined as the total expected discounted reward attained by taking action $a$, given current state $s$ and opponent action $o$, and then following the optimal policy from then on. It satisfies the following Bellman optimality equation\(^{23}\)

$$Q_n^{(m)}(s, a, o) \triangleq \mathbb{E}_{R, S'}[R(s, a, o) + \beta V_n^{(m)}(S')] \quad (5.1)$$

where the value function of a state $s$ for the LS is defined based on the minimax principle as

$$V_n^{(m)}(s) \triangleq \max_{\pi(s)} \min_{o} \sum_a Q_n^{(m)}(s, a, o) \pi(s, a) \quad (5.2)$$

with $\pi(s, a)$ denoting the probability of taking action $a$ at state $s$ under policy $\pi$. Based on the quality function, the minimax optimal policy for the LS at each state $s$ can be found by

$$\pi_n^{(m)}(s) = \arg \max_{\pi(s)} \min_{o} \sum_a Q_n^{(m)}(s, a, o) \pi(s, a). \quad (5.3)$$

The corresponding quantities of the attacker are defined similarly by replacing $R$ and $\pi$ with $R^O$ and $\pi^O$, respectively, and switching $a$ and $o$ in the above expressions.

The minimax-Q algorithm enables the LS and attacker to learn the optimal quality functions and policies gradually. In particular, after the $n$-th round of interaction, the LS updates its

\(^{23}\)In this work, superscripts $(m)$, $(w)$, $(mp)$ and $(wp)$ will be used to distinguish similar quantities in the conventional minimax-Q and WoLF, and the proposed minimax-PDS and WoLF-PDS algorithms, respectively.

\(^{3}\)In (6.2), $\mathbb{E}_{R, S'}$ denotes the expectation over the reward and the future state, with the capital letters $R$ and $S'$ denoting corresponding random variables. Similar notations will be used throughout this work.
quality function with a learning rate $\alpha_n \in (0, 1)$ by

$$
Q_{n+1}^{(m)}(s, a, o) = \begin{cases} 
(1 - \alpha_n)Q_n^{(m)}(s, a, o) + \alpha_n[r(s, a, o) + \beta V_n^{(m)}(s_{n+1})], & \text{for } (s, a, o) = (s_n, a_n, o_n), \\
Q_n^{(m)}(s, a, o), & \text{otherwise}.
\end{cases}
$$

(5.4)

The corresponding updated value function $V_{n+1}^{(m)}$ and policy $\pi_{n+1}^{(m)}$ can be computed by replacing $Q_n^{(m)}$ with $Q_{n+1}^{(m)}$ in the right hand side (RHS) of (6.3) and (5.3), respectively. The learning procedure of the attacker is similar. This minimax-Q iteration ensures that $Q_n^{(m)}(\pi_n^{(m)})$ converges to the $Q^{\ast}(\pi^{\ast})$ of the SG; but the minimax principle in (5.3) is conservative and may misguide the agent to blindly learn the worst case policy and cause performance loss. Intuitively, this may be best illustrated using the example given in [108]: Consider an opponent in a rock-paper-scissors game playing almost exclusively Rock, but playing Paper and Scissors with some small probability. Minimax-Q will converge to an equilibrium solution which randomizes among each of its three actions equally likely, but this is not a best response (playing only Paper in this situation is the only best response). Formally, this characteristic is known as irrationality, and the corresponding definition is given below [33, 108].

**Definition 4.** A MARL algorithm is said to be **rational** if it converges to a best response (i.e., an optimal policy that maximizes the expected reward) when the opponent plays a stationary policy (i.e., a policy that does not change over time).

To overcome the irrationality issue, a rational MARL algorithm, called Win-or-Learn Fast (WoLF), was developed in [108], with the penalty of losing convergence assurance. As indicated by its name, the WoLF algorithm updates the policy using a slow (fast) learning parameter $\delta_{\text{win}}$ ($\delta_{\text{lose}}$) when winning (losing). In particular, it is assumed that the agent is winning (losing) when its current policy provides larger (smaller) expected reward than an empirical average policy (defined later in Section 5.2.2). In addition, the optimal quality and value functions for the LS in the WoLF algorithm are defined as

$$
Q^{(w)}_n(s, a) \triangleq \mathbb{E}_{O,R,S'} \left[ R(s, a, O) + \beta \cdot V^{(w)}_n(S') \right],
$$

(5.5)

$$
V^{(w)}_n(s) \triangleq \max_a Q^{(w)}_n(s, a).
$$

(5.6)

The corresponding quantities for the attacker can be obtained similarly by switching the roles of the two.

It can be observed from (5.5) (with expectation over $O$) that, the WoLF algorithm includes the average effect of the opponent’s current action into the reward, instead of the presumed
worst-case effect from the opponent as in minimax-Q. The corresponding quality function can be updated through learning as

$$Q_{n+1}(s, a) = \begin{cases} (1 - \alpha_n)Q_n(s, a) + \alpha_n[r(s, a, o_n) + \beta V_n(s_{n+1})], & \text{for } (s, a) = (s_n, a_n), \\ Q_n(s, a), & \text{otherwise}, \end{cases}$$

while the updated value function $V_{n+1}^{(w)}$ can be obtained by replacing $Q^*$ with $Q_{n+1}^{(w)}$ in the RHS of (5.6).

### 5.1.3 Limitation of Conventional MARL

As will be exemplified in Section 5.3, the LS often holds certain extra information (e.g., statistics of its own communication and computational resources and schedules) as compared to the attacker in practical attack/defense games. However, none of the existing MARL algorithms can exploit such information advantage for performance improvement, even though it is highly desirable.

### 5.2 Multi-agent PDS Learning

In this section, two multi-agent post-decision state (PDS) learning algorithms, termed minimax-PDS and WoLF-PDS, are developed to enable an agent to learn and adapt faster in unknown dynamic environments when extra partial information is available. The analysis is given in the context of a general SG, and specific applications will be discussed in Section 5.3.

In the proposed multi-agent PDS-learning, it is assumed that after taking action $a$ at state $s$ given the opponent’s action $o$, the PDS-learning agent (i.e., the agent having extra information) will first receive a known reward $r^k(s, a, o)$ and the state transits to an intermediate state $\tilde{s}$, termed post-decision state, with a known probability $p^k(\tilde{s}|s, a, o)$, which is unknown to the opponent; then the state further transits to a future state $s'$ with an unknown probability $p^u(s'|\tilde{s}, a, o)$, and a reward $r^u(\tilde{s}, a, o)$ that depends on the random PDS $\tilde{s}$ is received. This process is illustrated in Fig. 5.2. Mathematically, the transition from $s$ to $s'$ admits

$$p(s'|s, a, o) = \sum_{\tilde{s}} p^u(s'|\tilde{s}, a, o)p^k(\tilde{s}|s, a, o),$$

Figure 5.2: Multi-agent PDS-learning.
and it can be verified that the expected reward of the state-action pair \((s, a, o)\) is given by
\[
E_\tilde{S}[R(s, a, o)] = r^k(s, a, o) + \sum_{\tilde{s}} p^k(\tilde{s}|s, a, o) r^u(\tilde{s}, a, o). 
\] (5.9)

### 5.2.1 Minimax-PDS

The basic idea of the proposed minimax-PDS is that, instead of directly updating a single quality function at a time, an agent with the extra information on \(p^k(\tilde{s}|s, a, o)\) and \(r^k(s, a, o)\) can first learn the PDS quality function \(\tilde{Q}^{(mp)}_*\) defined below and then simultaneously update multiple quality functions.

In particular, the optimal PDS quality function \(\tilde{Q}^{(mp)}_*\) for the post-decision state-action pair \((\tilde{s}, a, o)\) is defined as
\[
\tilde{Q}^{(mp)}_*(\tilde{s}, a, o) \triangleq r^u(\tilde{s}, a, o) + \beta \sum_{s'} p^u(s'|\tilde{s}, a, o) V^{(mp)}_*(s'), 
\] (5.10)
and the optimal quality and value functions \(Q^{(mp)}_*\) and \(V^{(mp)}_*\) in the proposed minimax-PDS are identical to \(Q^{(*)}_*\) and \(V^{(*)}_*\) defined in (6.2) and (6.3), respectively. Using the extra information \(p^k(\tilde{s}|s, a, o)\) and \(r^k(s, a, o)\), it can be shown (c.f. Appendix D.1) that \(Q^{(mp)}_*\) can be further expanded, for all state-action pairs \((s, a, o)\), as
\[
Q^{(mp)}_*(s, a, o) = r^k(s, a, o) + \sum_{\tilde{s}} p^k(\tilde{s}|s, a, o) \tilde{Q}^{(mp)}_*(\tilde{s}, a, o). 
\] (5.11)

In the proposed minimax-PDS algorithm, after observing the sample \((s_n, a_n, o_n, r^k(s_n, a_n, o_n), \tilde{s}_n, r^u(\tilde{s}_n, a_n, o_n), s_{n+1})\), the PDS-learning agent first updates the PDS quality function \(\tilde{Q}^{(mp)}_n\) by
\[
\tilde{Q}^{(mp)}_{n+1}(\tilde{s}_n, a_n, o_n) = (1 - \alpha_n) \tilde{Q}^{(mp)}_n(\tilde{s}_n, a_n, o_n) + \alpha_n [r^u(\tilde{s}_n, a_n, o_n) + \beta \cdot V^{(mp)}_n(s_{n+1})], 
\] (5.12)
and \(\tilde{Q}^{(mp)}_n(\tilde{s}, a, o) = \tilde{Q}^{(mp)}_n(\tilde{s}, a, o)\) for \((\tilde{s}, a, o) \neq (\tilde{s}_n, a_n, o_n)\).

After obtaining \(\tilde{Q}^{(mp)}_n, Q^{(mp)}_n\) can be updated using the RHS of (5.11) by replacing \(\tilde{Q}^{(mp)}_n\) with \(\tilde{Q}^{(mp)}_{n+1}\). Note that in the conventional minimax-Q, at each interaction, only a single entry \((s_n, a_n, o_n)\) of the quality function is updated using (6.4); while in contrast, all entries are updated in (5.11), and hence the learning speed is substantially accelerated. The corresponding \(V^{(mp)}_n\) and \(\pi^{(mp)}_{n+1}\) can be updated by replacing \(Q^{(*)}_*\) with \(Q^{(mp)}_{n+1}\) in the RHS of (6.3) and (5.3), respectively, for all states \(s\).

The convergence property of the proposed minimax-PDS is given by the following proposition.
Proposition 9. Using minimax-PDS, \( Q_{m}^{(mp)} \) converges to the minimax optimal quality function \( Q_{m}^{(mp)} \) with probability 1 when the learning rate sequence \( \alpha_n \) satisfies the conditions \( \alpha_n \in [0, 1) \), \( \sum_{n=0}^{\infty} \alpha_n = \infty \) and \( \sum_{n=0}^{\infty} \alpha_n^2 < \infty \).

Proof. Please see Appendix D.2.

However, as will be shown in Section 5.4, the minimax-PDS inherits the unfavorable irrationality property from the minimax-Q. To address this, a rational multi-agent PDS learning algorithm, termed WoLF-PDS, is developed in the next subsection.

5.2.2 WoLF-PDS

The WoLF-PDS is developed by incorporating the PDS-learning principle into the WoLF algorithm. The optimal quality and value functions in the proposed WoLF-PDS \( Q_{*}^{(wp)} \) and \( V_{*}^{(wp)} \) are defined identically to \( Q_{*}^{(w)} \) and \( V_{*}^{(w)} \), respectively, as in (5.5) and (5.6). The PDS quality function \( \tilde{Q}_{*}^{(wp)}(\tilde{s}, a) \) in the WoLF-PDS algorithm is defined as

\[
\tilde{Q}_{*}^{(wp)}(\tilde{s}, a) \triangleq \mathbb{E}_O \left[ r^{u}(\tilde{s}, a, O) + \beta \sum_{s'} p^{u}(s'|\tilde{s}, a, O)V_{*}^{(wp)}(s') \right].
\] (5.13)

Using the extra information, it can be shown (c.f. Appendix D.3) that \( Q_{*}^{(wp)}(s, a) \) can be further expanded as

\[
Q_{*}^{(wp)}(s, a) = \mathbb{E}_O[r^{k}(s, a, O)] + \sum_{\tilde{s}} p^{k}(\tilde{s}|s, a)\tilde{Q}_{*}^{(wp)}(\tilde{s}, a),
\] (5.14)

when the following assumption holds

\[
p^{k}(\tilde{s}|s, a, o) = p^{k}(\tilde{s}|s, a),
\] (5.15)

which requires that the transition from current state \( s \) to the PDS \( \tilde{s} \) is independent of the opponent’s action \( o \).

In the proposed WoLF-PDS learning, after observing the sample \( (s_n, a_n, o_n, r^{k}(s_n, a_n, o_n), \tilde{s}_n, r^{u}(\tilde{s}_n, a_n, o_n), s_{n+1}) \), the agent first updates the PDS quality function \( \tilde{Q}_{*}^{(wp)}(\tilde{s}, a) \) using

\[
\tilde{Q}_{n+1}^{(wp)}(\tilde{s}_n, a_n) = (1 - \alpha_n)\tilde{Q}_{n}^{(wp)}(\tilde{s}_n, a_n) + \alpha_n [r^{u}(\tilde{s}_n, a_n, o_n) + \beta \cdot V_{n}^{(wp)}(s_{n+1})],
\] (5.16)

and \( \tilde{Q}_{n+1}^{(wp)}(\tilde{s}, a) = \tilde{Q}_{*}^{(wp)}(\tilde{s}, a) \) for \( (\tilde{s}, a) \neq (\tilde{s}_n, a_n) \). In addition, it updates an empirical reward function \( \tilde{r}^{k} \) using

\[
\tilde{r}_{n+1}^{k}(s_n, a) = (1 - \alpha_n)\tilde{r}_{n}^{k}(s_n, a) + \alpha_n \cdot r^{k}(s_n, a, o_n),
\] (5.17)

Note that these requirements on \( \alpha_n \) appear in most of the RL algorithms (e.g., MDP, minimax-Q and WoLF) and are not specific to the proposed algorithms.
for all actions \(a\), and \(\tilde{r}_{n+1}^k(s, a) = \tilde{r}_n^k(s, a)\) for \(s \neq s_n\), so as to keep track of the empirical average performance. Then, the corresponding quality function \(Q_{n+1}^{(wp)}\) is updated by replacing \(E_O[r^k(s, a, O)]\) and \(Q_{n+1}^{(wp)}(\tilde{s}, a)\) in the RHS of (5.14) with \(\tilde{r}_{n+1}^k(s, a)\) and \(\tilde{Q}_{n+1}^{(wp)}(\tilde{s}, a)\), respectively, for all state-action pairs \((s, a)\).

After obtaining the quality functions, the rest steps of WoLF-PDS are similar to the original WoLF algorithm. In particular, the state occurrence count \(c\), the empirical average policy \(\bar{\pi}\) and the policy \(\pi^{(wp)}\) (with initial values given in Algorithm 5) are updated, respectively, as

\[
c_{n+1}(s_n) = c_n(s_n) + 1, \quad (5.18)
\]

\[
\bar{\pi}_{n+1}(s_n, a) = \bar{\pi}_n(s_n, a) + \frac{\pi^{(wp)}_n(s_n, a) - \bar{\pi}_n(s_n, a)}{c_{n+1}(s_n)}, \quad \forall a. \quad (5.19)
\]

\[
\pi^{(wp)}_{n+1}(s_n, a) = \pi^{(wp)}_n(s_n, a) + \Delta_{sa}, \quad \forall a, \quad (5.20)
\]

where \(\Delta_{sa} = -\delta_{sa}\) if \(a \neq \arg \max_a Q^{(wp)}(s, a')\) and \(\Delta_{sa} = \sum_{a 
eq a'} \delta_{sa'}\) otherwise, with \(\delta_{sa} = \min\{\pi^{(wp)}_n(s_n, a), \frac{\delta}{|A| - 1}\}\) and \(|A|\) denoting the size of the LS’s action space \(A\); intuitively, (5.20) moves the policy towards the highest valued action with speed controlled by the parameter \(\delta\). At each round, \(\delta\) is determined by the WoLF principle. Particularly, \(\delta = \delta_{\text{win}}\) if \(\sum_{a'} \pi^{(wp)}_n(s_n, a')Q_{n+1}^{(wp)}(s_n, a') > \sum_{a'} \bar{\pi}_n(s_n, a')Q_{n+1}^{(wp)}(s_n, a')\) (the winning condition) and \(\delta = \delta_{\text{lose}} (> \delta_{\text{win}})\) otherwise, where both \(\delta_{\text{win}}\) and \(\delta_{\text{lose}}\) vanish over (usually inversely proportionally to) time [108].

Through updating multiple \(Q^{(wp)}\)'s at a time by taking advantage of extra information, the proposed WoLF-PDS algorithm can substantially expedite the learning speed as compared to the original WoLF algorithm. In addition, it admits the favorable rationality property.

**Proposition 10.** The WoLF-PDS is rational when the sequence \(\alpha_n\) satisfies the conditions in Proposition 9.

**Proof.** Please see Appendix D.4. \(\square\)

Nonetheless, similar to the conventional WoLF algorithm, the proposed WoLF-PDS has no convergence assurance in general SG.

In practice, an agent has to switch between taking actions uniformly at random (with probability \(p_{\text{explor}}\)) and following the learned policy (with probability \(1 - p_{\text{explor}}\)), to ensure sufficient explorations of the underlying SG [33]. The corresponding algorithms of the proposed minimax-PDS and WoLF-PDS are given in Algorithm 4 and Algorithm 5, respectively.
**Algorithm 4** Minimax-PDS algorithm.

Initialization: \( n = 1, Q^{(mp)} = 0, V^{(mp)} = 0 \) and \( \pi^{(mp)} \) uniform.

Taking action \( a_n \) at current state \( s_n \)

- uniformly at random with probability \( p_{explor} \);
- otherwise, with probability \( \pi_n^{(mp)}(s_n, a_n) \).

Learning: after receiving a reward \( r_k(s_n, a_n, o_n) \) and observing the state transition from \( s_n \) to \( \tilde{s}_n \) and then to \( s_{n+1} \)

- Update \( Q^{(mp)} \) using (5.12);
- Update \( \tilde{Q}^{(mp)} \) using (5.11) (with \( \tilde{Q}^{(mp)}_s \) replaced by the updated \( \tilde{Q}^{(mp)} \));
- Update \( V^{(mp)} \) and \( \pi^{(mp)} \) using (6.3) and (5.3) (with \( Q^{(m)}_s \) replaced by the updated \( Q^{(mp)} \)), respectively.

Repeat.

**Algorithm 5** WoLF-PDS algorithm.

Initialization: \( n = 1, Q^{(wp)} = 0, c = 0, \bar{r} = 0, \) and \( \pi^{(wp)} \) and \( \bar{\pi} \) uniform.

Taking action \( a_n \) at current state \( s_n \)

- uniformly at random with probability \( p_{explor} \);
- otherwise, with probability \( \pi_n^{(wp)}(s_n, a_n) \).

Learning: after receiving a reward \( r_k(s_n, a_n, o_n) \) and observing the state transition from \( s_n \) to \( \tilde{s}_n \) and then to \( s_{n+1} \)

- Update \( \tilde{Q}^{(wp)} \) using (5.16);
- Update \( \bar{r}^k \) using (5.17);
- Update \( Q^{(wp)} \) using (5.14) (with \( \mathbb{E}_O[r_k(s, a, O)] \) and \( \tilde{Q}^{(wp)}_s \) replaced by the updated \( \bar{r}^k \) and \( Q^{(wp)} \), respectively);
- Update \( c \) using (5.18);
- Update \( \bar{\pi} \) and \( \pi^{(wp)} \) using (5.19) and (5.20), respectively.

Repeat.
5.3 Applications in Practical Anti-jamming Games

Three applications of the proposed algorithms are presented in this section to provide concrete instances for the general framework presented above. EH and CR technologies are promising for future communication networks as they improve the utilization efficiency of the scarce energy and spectrum resources, respectively. The first two examples illustrate how EHCS and CR systems can employ the proposed algorithms as effective defense against jamming attacks. Cloud based security service [109, 110] is another emerging application in which users can utilize cloud resource to fulfill various resource-demanding security applications (e.g., intrusion detection [111, 112]). The third example demonstrates how the proposed algorithms can guide a cloud user to strategically utilize the dynamic cloud resource so as to conduct most effective security defense.

5.3.1 Anti-jamming in EHCS

The first example considers the jamming/anti-jamming competition between an EHCS and a jammer (J) in a dynamic environment, where the channel power gain $h_n \in \mathcal{H} \triangleq \{h^{(1)}, ..., h^{(m)}\}$ varies over time following an Markov process with transition probability $p(h_{n+1} = h^{(j)}|h_n = h^{(i)}) = p_H(j|i)$ unknown to both the EHCS and the jammer. As illustrated in Fig. 5.3, the competition proceeds as follows: At each timeslot $n$, the EHCS chooses a suitable transmit power $a_n$, which is constrained by its current battery energy level $b_n \in \mathcal{B} \triangleq \{b^{(1)}, ..., b^{\text{max}}\}$ with $b^{(1)} = 0$ and $b^{\text{max}}$ the battery capacity. Without loss of generality, it is assumed that $\delta b$ amount of new energy will be harvested by the EHCS after each transmission (with duration $T_s$), transiting its battery level from $b_n - a_n T_s$ to $b_{n+1} = \min\{b_n - a_n T_s + \delta b, b^{\text{max}}\}$ with probability $p_{EH}(b_{n+1}|b_n - a_n T_s)$. In addition, we assume that $p_{EH}$ is known only to the EHCS itself, inducing information asymmetry. On the other hand, due to a per unit jamming power cost $c_J$, the smart jammer needs to select a proper jamming power $o_n \in \mathcal{O}$ for effective jamming. Considering the zero-sum assumption, the total reward of the EHCS at each timeslot $n$ is modeled as the sum of the throughput reward and the cost of the jammer, and is given by
\[ r_n = \frac{h_n \cdot a_n}{(o_n + N)} + c_j \cdot o_n, \quad (5.21) \]

\[ = \text{signal-to-interference-plus-noise ratio} \]

\[ \text{jamming cost} \]

where \( N \) is the noise power and a unit jamming channel power gain is assumed for simplicity. The jammer's reward is the opposite.

### 5.3.2 Anti-jamming in CR Systems

The second example envisages the competition between a SU and a primary user emulation attacker (PUEA), as shown in Fig. 5.4. In the PUE attack, the adversary emits emulated PU signals to spoof SUs, aiming at obtaining exclusive access of the spectrum holes [113], and may be considered as a smart-type jammer to CR systems. Assume that the on/off state of the PU is Markovian, and the probability \( \phi_{1,0} \) (\( \phi_{0,1} \)) that the PU (e.g., a TV tower) transits from the active (inactive) to inactive (active) state is known to both the SU and the PUEA [17].

The PU spectrum is divided into \( k \) sub-channels, and the channel power gain \( h_i \) of each sub-channel \( i \) independently varies among \( m \) different states \( \mathcal{H}_i = \{ h^{i,1}, ..., h^{i,m} \} \) in a Markovian manner with unknown statistics \( p_{hi} \) to both the SU and the PUEA; also note that all the \( k \) sub-channels will be occupied when PU is active. At each timeslot \( n \), a sensing token \( 1_S^S \in \{0, 1\} \) indicating whether the SU can sense/transmit or not will be assigned to the SU, which captures, for example, the random access scenario in a CSMA based secondary network; the transition probability \( p_S(1_S^S|1_{S,n-1}^S) \) of the sensing token is assumed known only to the SU. If an inactive sensing token \( 1_S^S = 0 \) is obtained, the SU will keep silent, denoted by \( a_n = 0 \), otherwise it will choose a target channel \( i \in \{0, 1, ..., k\} \) for sensing, denoted by \( a_n = i \) (including the option of keeping silent). If no PU signal is detected on the sensed channel, the SU will transmit on this channel, given that the current data queue length \( q_n > 0 \). At the same time, the PUEA will also either choose a target channel \( i' \) to emit faked PU signals, denoted by \( o_n = i' \), or keep silent (to avoid a penalty \( c_P \) caused by the collision of the real and the faked PU signals), denoted by \( o_n = 0 \). For simplicity, we assume that spectrum sensing is perfect but the SU cannot distinguish between the real and the faked PU signals. Also note that PUE attack must be launched at the SU sensing period to be effective. In each timeslot, \( \Delta q_n = 1_S^S \cdot \sum_{i=1}^{k} 1_{\{a_n = i, o_n \neq i\}} \cdot \min\{q_n, C(h_{ni}^i)\} \) amount of data can be successfully transmitted by the SU, where \( 1_{\{\cdot\}} \in \{0, 1\} \) is the indicator function and \( C(h_{ni}^i) = B_i \log(1 + h_{ni}^i \cdot P/N_i) \) is the capacity of the \( i \)th sub-channel with channel power gain \( h_{ni}^i \in \mathcal{H}_i \), bandwidth \( B_i \), transmit power \( P \) and noise power \( N_i \). In addition, \( \delta q \) units of new data will arrive at the SU, transiting the data queue state to \( q_{n+1} = q_n - \Delta q_n + \delta q \) with probability \( p_D(q_{n+1}|q_n - \Delta q_n) \), assumed known only to the SU, which, together with the knowledge on the sensing token transition probability \( p_S \), induces information asymmetry. The
expected reward of the SU is modeled as

\[ r_n = \phi_{PU_{n-1}} \cdot \Delta q_n - 1^n \cdot 1_{\{a_n>0\}} \cdot c_s + c_p \cdot 1_{\{o_n>0\}} \cdot \phi_{PU_{n-1}}, \]  

(5.22)

where \( c_s \) is the sensing cost (e.g., due to circuit power consumption) of the SU; \( 1^{PU_{n-1}} \in \{0, 1\} \) is the PU state in the previous timeslot \( n - 1 \) which is assumed known to both the SU and the PUEA. The PUEA’s reward is the opposite.

5.3.3 Cloud-based Security Game

The third example considers a cloud-based security game as depicted in Fig. 5.5. At each timeslot \( n \), the defender (D) and the attacker (A) choose \( a_n \) and \( o_n \) nodes in the target network to enforce security protection and to inject malware, respectively. To fulfill the security protection to these \( a_n \) nodes, the defender needs to request \( a_n \) units of computing resource from the cloud. As in [114], it is assumed in this work that the cloud provides both exclusive and opportunistic resources; the exclusive resource has a per unit price \( \varphi_a \) while the opportunistic resource is
free and its availability follows a Markov process with transition probabilities $p_C$, which is assumed known only to the defender and induces information asymmetry. Denoting by $c_n \in C = \{0, \ldots, c_{\text{max}}\}$ the amount of opportunistic resource available at timeslot $n$, the payment from the defender will be $\phi_a \cdot \max\{a_n - c_n, 0\}$. While for the attacker, a per node attacking cost $\phi_o$ is assumed. In this work, it is assumed that each of the $K$ nodes in the target network can be in either a healthy or an infected state. In addition, it is assumed that when both the defender and the attacker act on the same node, this node will transit from infected (healthy) state to healthy (infected) state with probability $p_{10}$ ($p_{01}$), which is unknown to both the defender and attacker\(^5\); when only the defender (attacker) acts on a node, this node will result in a healthy (infected) state; otherwise, the state of the node remains unchanged. Considering the possibility of malware spreading, it is further assumed that, after both the attacker and the defender take actions, a spreading phase will occur, in which, any healthy node may be infected by a infected node with unknown probability $p_{\text{inf}}$\(^6\). In this security game, the defender aims at maximizing the number of healthy nodes with minimum payment and its instant reward at timeslot $n$ is given by

$$r_n = (K - k_n) - \phi_a \cdot \max\{a_n - c_n, 0\} + \phi_o \cdot o_n,$$

(5.23)

where $k_n$ denotes the number of infected nodes at timeslot $n$. The reward of the attacker is assumed to be the opposite.

### 5.3.4 Applications of the Proposed Algorithms

To apply the proposed algorithms, the EHCS anti-jamming problem is first formulated as a SG with the state defined as $s_n = (b_n, h_n)$ and the corresponding actions and rewards defined as in Section 5.3.1. In addition, the PDS can be defined as $s_n = (b_n, h_n)$ with $b_n \triangleq \min\{b_n - a_n T_s + \delta b, b_{\text{max}}\} = b_{n+1}$, which is the system state after transmission and arrival of new energy. With this PDS definition, the corresponding known and unknown state transition probabilities are $p^k = p_{EH}$ and $p^u = p_H$, respectively; the known reward $r^k_n$ is given by (5.21) and $r^u_n = 0$.

For the SU anti-jamming problem, the state is defined as $s_n = (h_n, q_n, 1^{(\text{PU})}_{n-1}^{\text{p}}, 1^S_n)$, and the PDS is defined as $s_n = (h_n, q_n, 1_n^{(\text{PU})^{p_m}}, 1_n^S)$ with $q_n \triangleq q_n - \Delta q_n + \delta q = q_{n+1}$ and $1_n^S \triangleq 1^S_{n+1}$, which is the system state after the data transmission and the arrivals of new data and sensing token; accordingly, $p^k = p_D \cdot p_S$ and $p^u = \prod_{i=1}^k p_{h_i}$, and $r^k_n$ is given by (5.22) and $r^u_n = 0$. Note that, since the transition from $s_n$ to $s_n$ depends on the PUEA’s action (which affects the successfulness of a SU transmission and its data queue state), (5.15) is violated.

\(^5\)It is usually difficult to predict $p_{10}$ and $p_{01}$ beforehand, since they may depend on the effectiveness of the specific defense to the malware.

\(^6\)In practice, $p_{\text{inf}}$ is determined by the properties of both the malware and the target network, and thus, neither the defender nor the attacker can predict $p_{\text{inf}}$ unilaterally.
Therefore, WoLF-PDS cannot be applied and only minimax-PDS will be considered in this SU anti-jamming problem.

For the cloud-based security game, the state is defined as $s_n = (1_1, ..., 1_K, c_n)$, where $1_i = 0$ if the i-th node is healthy and otherwise $1_i = 1$, and the corresponding PDS is defined as $\hat{s}_n = (1_1, ..., 1_K, \hat{c}_n)$ with $\hat{c}_n \triangleq c_{n+1}$. With these definitions, $p^k = p_C$, and $p^a$ is determined by the probabilities of node state transition $p_{01}$ and $p_{10}$ and the probability of malware spreading $p_{inf}$. Accordingly, the known part of the reward $r^k_n$ is given by (5.23) and $r^a_n = 0$.

In our setting, the jammer (in the EHCS and CR applications) and the attacker (in the cloud-based security game) cannot employ the proposed minimax-PDS or WoLF-PDS due to lack of knowledge about $p_{EH}$, $p_D$ and $p_S$, and $p_C$.\(^7\) Also, to focus on the main theme of this work, we have made some assumptions about the underlying problems above for simplicity; some of them will be addressed in Section 5.4 while the others are left to future work.

### 5.4 Simulations

In this section, numerical results are presented to justify the effectiveness of the proposed algorithms. Specifically, the performance gain of using the proposed algorithms is measured by

$$\eta(n) \triangleq \frac{\bar{r}_{PDS}(n) - \bar{r}(n)}{\bar{r}(n)} \times 100\%,$$

where $\bar{r}_{PDS}(n) \triangleq \frac{1}{n} \sum_{i=1}^{n} r(s_i, a_i, o_i)$ is the average accumulative reward [115] till timeslot $n$ when the LS adopts PDS-learning; $\bar{r}(n)$ is defined similarly when the LS adopts the conventional minimax-Q or WoLF algorithm; $\bar{r}(n)$ denotes the average of $\bar{r}(n)$ over all Monte Carlo runs and serves as a normalization factor here. For learning speed comparison, the relative distance between the learned and the optimal quality functions $Q_n$ and $Q^*$, defined as

$$\Delta Q_n \triangleq ||\text{vec}(Q_n - Q^*)||_1 / ||\text{vec}(Q^*)||_1 \times 100\%,$$

is the metric of interest, with $\text{vec}(\cdot)$ the vectorization operator and $|| \cdot ||_1$ the 1-norm.\(^8\) For minimax-Q and minimax-PDS the corresponding $Q^*$ can be found by numerically computing the fixed point of (6.2) and (6.3); for WoLF and WoLF-PDS, $Q^*$ in the stationary jammer case can be found by computing the fixed point of (5.5) and (5.6).\(^9\) The learning rates $\alpha_n$’s in

\(^7\)Although the jammer and the attacker may adopt model-learning approaches [33], in which an agent jointly estimates the unknown model and computes the strategy, it is beyond the scope of this work, since our main objective in this work is to improve the learning performance of model-free algorithms [33] (e.g., minimax-Q and WoLF) by exploiting information asymmetry. Also, model-learning approaches suffer from significant higher computation complexity (due to the costly fixed point evaluations at every interaction) as compared to the model-free ones (only using simple updates as (6.4)).

\(^8\)Note that in a finite-dimensional vector space, convergences in 1-norm and max-norm are equivalent, but, in our view, 1-norm is more natural for relative distance comparison here.

\(^9\)For WoLF and WoLF-PDS with non-stationary jammers, opponent independent $Q^*$ does not exist in general.
minimax-PDS and WoLF-PDS are set according to [107] and [108], respectively. All the average values are obtained through 100 Monte Carlo runs.

5.4.1 EHCS Anti-jamming

For the EHCS anti-jamming application, the following scenario (scenario I) is considered. Specifically, the EHCS battery level varies among \( \mathcal{B} = \{b^{(1)}, b^{(2)}, b^{(3)}\} = \{0, 5, 10\} \) and the corresponding transition probability matrix \( p_{EH} = \begin{bmatrix} 0.6 & 0.4 & 0 \\ 0 & 0.9 & 0.1 \end{bmatrix} \); the channel power gain changes between two states \( \mathcal{H} = \{h^{(1)}, h^{(2)}\} = \{1, 100\} \) with transition probability matrix \( p_{H} = \begin{bmatrix} 0.7 & 0.3 \\ 0.5 & 0.5 \end{bmatrix} \); the jammer chooses the jamming power from the action set \( \mathcal{O} = \{0, 5, 10\} \); \( p_{\text{explor}} \) and the discounting factor \( \beta \) are set to 0.2 and 0.75, respectively; the transmission duration \( T_s \) and noise power \( N \) are set to 1 and 0.5, respectively; the jammer power cost per unit \( c_J \) is set to 1.

**Minimax-PDS and WoLF-PDS**

The performance of the proposed minimax-PDS and the conventional minimax-Q when the EHCS faces a minimax-Q jammer is examined first. Fig. 5.6(a) shows the average performance gain \( \bar{\eta}(n) \) provided by the proposed minimax-PDS (curve Minimax-PDS (\( p_{FI} = 0 \))) over the conventional minimax-Q (curve Minimax-Q (\( p_{FI} = 0 \))), and Fig. 5.6(b) compares the learning speeds. It can be seen from Fig. 5.6(b) that the proposed minimax-PDS enables the EHCS to learn the minimax optimal policy substantially faster as compared to the conventional minimax-Q, which in turn leads to a significant average performance gain for the EHCS. For example, as shown in Fig. 5.6(b), even till \( n = 800 \), the average relative distance \( \Delta Q_n \) between \( Q_\star \) and the \( Q_n^{(m)} \) learned by the conventional minimax-Q algorithm is still above 50%; in contrast, the minimax-PDS favorably learns a \( Q_n^{(mp)} \) with average relative distance around 3% to \( Q_\star \), and as
a consequence, an average performance gain of 31% is exhibited in Fig. 5.6(a). In addition, it can be observed from Fig. 5.6(a) that minimax-PDS only benefits EHCS at the learning phase, and the corresponding gain decreases over time. The reason is that for fixed channel statistics $p_H$, both minimax-Q and minimax-PDS will eventually converge to the optimal policy and hence perform identically from then on. However, in practice, the channel statistics $p_H$ will also vary over time, forcing both the EHCS and the jammer to stay frequently in the learning phase. In such cases, the minimax-PDS is apparently more favorable. The corresponding gain may be coined as the \textit{hiding target defense gain}, as it is achieved by the uncertainty (hiding) of some information at the LS side.

In the above simulation, we give the jammer the privilege of perfectly observing the battery state $b_n$ of the EHCS, which could be true in applications where the EHCS has to report its battery state to a control center and the information is intercepted by the jammer. In practice, the jammer may fail to obtain such information or the EHCS can deliberately send out falsified information with certain probability $p_{FI}$. In such cases, assuming that the jammer will take a conservative estimate of $b_n$ as $b^{\text{max}}$ whenever it fails to intercept $b_n$, the corresponding performance curves for $p_{FI} = 0.2$ are also shown in Fig. 5.6(a). It can be seen that, a nonzero $p_{FI}$ leads to extra anti-jamming gains for both minimax-Q and minimax-PDS EHCSs, since the jammer’s learning process is disrupted by the false information. The advantage obtained through this type of information asymmetry may be termed \textit{falsifying target defense gain}. A more comprehensive examination of the impacts of such falsified information is beyond the scope of this work, and $p_{FI} = 0$ is assumed in the rest simulations for simplicity.

The performance of the WoLF-PDS EHCS against a WoLF jammer is shown in Fig. 5.7, and it can be seen that the proposed WoLF-PDS provides a significant anti-jamming performance gain (up to 92% at $n = 50$) to the EHCS as compared to the conventional WoLF. This can be explained by its capability of faster adaptation to the jammer. Since WoLF-PDS generally does not have convergence assurance against a dynamic opponent, e.g., the WoLF jammer considered here, its faster adaptation capability will be illustrated next, through stationary jammer examples.

Moreover, the proposed minimax-PDS also demonstrates similar advantages when facing a WoLF jammer, and so does the proposed WoLF-PDS when facing a minimax-Q jammer, as shown in Fig. 5.8 and Fig. 5.9, respectively. These results indicate that the effectiveness of the proposed algorithms does not depend on the opponent’s learning process.

\textbf{Convergence vs. Rationality}

Another scenario (scenario II) is deliberately chosen to discuss the convergence and rationality aspects of the proposed algorithms. Particularly, it is assumed that $p_{EH} = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.3 & 0.5 & 0.5 \\ 0 & 0 & 1 \end{bmatrix}$, $\mathcal{H} =$
\{h^{(1)}, h^{(2)} \} = \{0.5, 40\}, \; p_H = [0.7, 0.3], \; \beta = 0.3, \; \text{and other parameters remain the same as in scenario I. In addition, two stationary jammers, } sJ_1 \text{ and } sJ_2, \text{ are considered. Specifically, } sJ_1 \text{ adopts a stationary policy } \pi_{sJ_1}(s) = [1/3, 1/3, 1/3] \text{ at all states } s, \text{ i.e., taking a jamming action } o \text{ from } O \text{ with equal probability each time, while } sJ_2 \text{ adopts a more aggressive stationary policy } \pi_{sJ_2}(s) = [0.1, 0.1, 0.8] \text{ at all states } s.\]

The reward performance of the conventional and the proposed algorithms are compared in Fig. 5.10 when the EHCS faces } sJ_1 \text{ and } sJ_2, \text{ respectively. From Fig. 5.10, it can be observed that in both cases, the minimax-PDS does not provide as much anti-jamming gain as the WoLF-PDS, which is due to its } \text{irrationality} \text{ property inherited from the conventional minimax-Q. In fact, the minimax principle forces the EHCS to take a conservative policy (best response to the worst opponent strategy) even when the opponents (} sJ_1 \text{ and } sJ_2 \text{) are actually not playing with the worst case policy. Nonetheless, it is worth noting that the minimax-PDS converges to the same minimax optimal quality function } Q^{(mp)}_{s}(s) \text{ in both cases as shown in Fig. 5.11, as promised by Proposition 9. In addition, the minimax optimality indicates that the jammer cannot further degrade the performance, and the minimax-PDS allows the EHCS to obtain such a stable policy more quickly as compared to minimax-Q.}\n
Unlike the minimax-PDS that always converges to an opponent independent minimax optimal } Q^{(mp)}_{s}(s), \text{ it can be seen from the trends in Fig. 5.12(a) and Fig. 5.12(b) that, the rational (c.f. Definition 4) WoLF-PDS will converge to two different } Q_{s}'s \text{ corresponding to the two dif-
ferent best responses against $sJ_1$ and $sJ_2$, respectively, as suggested by Proposition 10. (In Fig. 5.12(b), the optimal $Q_*$ against $sJ_1$ is out of the border.) In addition, the proposed WoLF-PDS offers substantially faster learning as compared to the conventional WoLF. When facing a dynamic jammer (as considered in scenario I), faster learning capability along with rationality allows more agile adaptation to the opponent’s current policy and thus leads to better LS performance.

5.4.2 SU Anti-jamming

For the SU anti-jamming application, it is assumed that the PU transition statistics are $\phi_{1,0} = \phi_{0,1} = 0.5$; the PU spectrum is divided into $k = 2$ sub-channels, and the first (second) channel varies between two power gains $\mathcal{H}^1 = \{h_1^{1,(1)}, h_1^{1,(2)}\}$ ($\mathcal{H}^2 = \{h_2^{2,(1)}, h_2^{2,(2)}\}$) with transition probabilities $p_{h_1^{1}} = \begin{bmatrix} 0.7 & 0.3 \\ 0.2 & 0.8 \end{bmatrix}$ ($p_{h_2^{1}} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$) and the corresponding channel capacities are $C(h_1^{1,(1)}) = 1$ and $C(h_1^{1,(2)}) = 2$ ($C(h_2^{2,(1)}) = 0$ and $C(h_2^{2,(2)}) = 1$); the data queue length $q_n \in \{0, 1, 2\}$ and corresponding transition matrix is $p_D = \begin{bmatrix} 0.1 & 0.8 & 0.1 \\ 0 & 0.2 & 0.8 \end{bmatrix}$; the sensing token transition matrix $p_S = \begin{bmatrix} 0.2 & 0.8 \\ 0.5 & 0.5 \end{bmatrix}$; and $c_s = 0.7$, $c_p = 2$, $p_{\text{explor}} = 0.2$, and $\beta = 0.75$.

The performance of the proposed minimax-PDS and the conventional minimax-Q at a SU when facing a minimax-Q PUEA is compared in Fig. 5.13, similar to what is presented in Fig. 5.6 for the EHCS application. It can be seen from Fig. 5.13(b) that the SU learns the optimal policy much faster by using minimax-PDS, which in turn, results in a substantial performance
gain (up to 45%) in rewards in the learning phase, as shown in Fig. 5.13(a). In addition, similar observation can be made as in the EHCS case when the SU sends out falsified information (with $p_{FI} = 0.2$).

### 5.4.3 Cloud-based Security Game

For the cloud-based security game, it is assumed that there are $K = 3$ nodes in the target network; the per node attacking cost is $\varphi_o = 0.5$ and the per unit exclusive cloud resource price is $\varphi_a = 1.5$; the probabilities of the node state transition and malware spreading are set to $p_{01} = p_{10} = 0.3$ and $p_{inf} = 0.5$, respectively; the available opportunistic cloud resources change among the values $C = \{0, 1, 2\}$ with transition probabilities $p_C = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$; $p_{explor}$ and $\beta$ are set to 0.75 and 0.2, respectively. Note that, in the cloud-based security game, it is reasonable to assumed that both the node states and the amount of available opportunistic cloud resource are publicly known, and thus we set $p_{FI} = 0$ in this example.

Similar to the previous two applications, the proposed minimax-PDS and WoLF-PDS again
demonstrate significant performance gain over the conventional minimax-Q and WoLF algorithms, as shown in Fig. 5.14 and Fig. 5.15 respectively.

5.5 Related Works

Game theory has been widely adopted in the network security study ([29, 30] and references therein), and both the simultaneous-move game model (e.g., [31]) and Stackelberg game model (e.g., [32]) have been considered; but the majority only considers non-stochastic games. Security games with incomplete information receive research interest recently ([116] and references therein), where it is shown that incomplete information about the opponent will usually lead to performance degradation in jamming/anti-jamming games. However, the classical game theory framework adopted by these works often assumes a known environment and each player purely focuses on how to deal with the opponent optimally through a Bayesian approach, which does not fit well in unknown dynamic environments. Our work addresses the incomplete information issue from a new angle, i.e., how extra local information may be exploited by the player to gain advantage over the opponent, in the context of SGs with unknown environmental dynamics.

Among the existing works adopting SG formulation and MARL, [115] and [117] are the most
relevant to our work, where the minimax-Q and the WoLF algorithms were applied, respectively, to find anti-jamming strategies in multi-channel CR systems. In [115] and [117], the LS and the attacker are treated as two equally knowledgeable learning agents. When the LS lacks sufficient knowledge about the opponent, existing works either impose restrictions to the considered opponent models (e.g., [118, 119]) or treat the opponent and the dynamic environment as an integrated entity (e.g., [120]) and then further invoke MDP or multi-armed bandit (MAB) algorithms for strategy learning; the former approach may fail in the presence of an intelligent opponent, while the performance of the latter is only comparable to a static forecaster and hence may not be sufficiently good in practice. Our work is orthogonal to these works and focuses on exploiting local information unknown to the opponent for performance improvement. In addition, our algorithms advance the pioneer work of single agent PDS-learning [121], which expedites system learning speed in benign environments, to the more complicated multi-agent cases so that they can deal with intelligent adversaries.

Moreover, the research on EHCS security and cloud-based security systems is still in the infancy, as compared to that on CR security [122]. Some pioneer works in EHCS security include the study of the secrecy data rate in [123] and [124], energy harvesting friendly jamming in [125], and the impact of energy harvesting on smart meter privacy in [126]; but there exists little work that can exploit unique features of the EHCS for anti-jamming performance enhancement, and our work contributes in this direction. As to the cloud-based security systems, existing works mainly focus on architectural designs and (proof-of-concept) implementations [109–112], and our work is the first to consider security game in such systems.

5.6 Conclusion

Two new MARL algorithms, minimax-PDS and WoLF-PDS, that enable a game player to learn and adapt faster in unknown dynamic environments by exploiting information asymmetry between itself and the opponent, are proposed in this work, which are provably convergent and rational, respectively. Numerical results verify that the proposed algorithms can provide substantial performance improvement as compared to conventional ones. The proposed learning framework is general enough to admit wide applications.
Chapter 6

A Stochastic Multi-channel Spectrum Access Game with Incomplete Information

In the previous chapter, security games with information advantage at the defender side are investigated; while in this chapter, the complementary problem of how to handle the incomplete information about the opponent is explored. Particularly, the zero-sum incomplete information SG is studied in the context of wireless communication networks.

Although the SG framework and RL [33] is suitable to capture the unknown dynamics in the wireless environments and enable wireless users to gradually learn proper communication strategies. Conventional RL algorithms usually target solving SG with complete information, where player types (e.g., friendly or malicious) are assumed perfectly known; while in real games, especially those involved with human or any other intelligent entity, a player may not always have precise information about and sometimes may even be deceived by other peers. In a wireless communication network, malicious users may disguise themselves using faked ID or through compromising legitimate users. In such cases, always presuming a malicious peer may be too conservative as it will apparently incur performance loss when the other player is actually not. A more proper way is to model such scenarios as incomplete information SG, in which a player needs to handle both the unknown environment dynamics and uncertainty in other players’ types. Solving a general incomplete information SG is challenging, and to the best of our knowledge, none of the existing RL algorithms can be directly applied.

In this chapter, a two-player multi-channel spectrum access game (SAG) is considered as a concrete example of incomplete information SG, where the information incompleteness stems from player-I’s ambiguity in player-II’s type (friend or enemy). For this SAG, a joint RL and type identification algorithm is proposed to find the corresponding best strategy for player-I,
with the understanding that actions of a player are type-driven and hence can be exploited for type detection. Through embedding a proper type detection procedure into the strategy learning process, a player employing the proposed algorithm can eventually achieve the same performance as that in the corresponding game with complete information.

The remainder of this chapter is organized as follows. Section 6.1 introduces SG and RL, and formulates the SAG in the context of SG. The proposed algorithm is presented in Section 6.2 and its performance in SAG is examined numerically in Section 6.3. Related works are discussed in Section 6.4. Section 6.5 concludes this chapter.

6.1 Problem Formulation and Background

This section starts with a brief review of SG, and then the SAGs with complete and incomplete information are presented, respectively, in the context of SG.

6.1.1 General Stochastic Game and the Spectrum Access Game

In general, a SG [33] between two players, as shown in Fig. 6.1, can be characterized by the tuple $\langle S, \mathcal{A}^I, \mathcal{A}^II, T, R^I, R^II \rangle$, where $S$ stands for the state space; $\mathcal{A}^I$ and $R^I$ denote the action space and the reward (also termed as payoff) function of player-I, respectively; $\mathcal{A}^II$ and $R^II$ denote those of player-II; and $T: S \times \mathcal{A}^I \times \mathcal{A}^II \rightarrow p(S)$ is the state transition function that maps the current state $s$ and the actions $(a^I, a^II)$ of the two players into a distribution $p(s'|s, a^I, a^II)$ of the future state $s'$. In a dynamic environment, both players conduct far-sighted optimization to learn strategies $\pi^I: S \rightarrow p(\mathcal{A}^I)$ and $\pi^II: S \rightarrow p(\mathcal{A}^II)$ so as to maximize their own cumulative discounted rewards $\mathbb{E}\{\sum_{n=1}^{\infty} \beta^n R^I_n\}$ and $\mathbb{E}\{\sum_{n=1}^{\infty} \beta^n R^II_n\}$ (with discounting factor $\beta \in [0, 1]$), respectively, which represent their long-term performance with diminishing weighting on the future. At each timeslot $n$, both players observe the current state $s_n \in S$ and take actions $a^I_n \in \mathcal{A}^I$ and $a^II_n \in \mathcal{A}^II$ according to the learned strategies $\pi^I_n$ and $\pi^II_n$, respectively;\(^1\) then,

\(^1\)Perfect monitoring, i.e., both players can perfectly observe each other’s action, is assumed throughout this chapter.
they receive rewards \( r^I_n = R^I(s_n, a^I_n, a^{II}_n) \) and \( r^{II}_n = R^{II}(s_n, a^I_n, a^{II}_n) \), respectively. Meanwhile, the environment transits to a new state \( s_{n+1} \in \mathcal{S} \) according to the (often unknown) state transition function \( T \).

The above SG modeling is a good fit to the following SAG problem, which serves to provide the context for our study, but is also of interest in its own right. Particularly, a \( k \)-channel wireless communication system is considered in the SAG, and the channel power gains of all these \( k \) channels \( \mathcal{H} = \{h^i_{n}\}_{i=1}^k \) are assumed to vary over time with each following an independent and unknown Markov process. In light of the dynamic sub-channel allocation framework pioneered in [127] (where the base station dynamically assigns sub-channels to users according to the instant channel state information), in this chapter, it is assumed similarly that there is a system administrator (SA) constantly monitoring the channel usages and conducting dynamic channel allocation accordingly; to release the computation burden at the SA, it is further assumed that the SA does not meddle in the users’ decision-making processes and allows users themselves to autonomously learn their optimal utilization strategies among the assigned channels in a distributed manner. More specifically, at each timeslot \( n \), the SA assigns \( m \) out of \( k \) channels with indices \( I_n = \{i_1, ..., i_m\} \) to a group of users\(^2\). This chapter focuses on the case of two users, referred to as user-I and user-II respectively, as is common for a game-theoretic study; the more general multi-user case will be considered in future work. At each timeslot \( n \), after observing the state \( s_n \), defined as the power gains of the assigned channels \( h^i_{n, 1}, ..., h^i_{n, m} \), both users will choose a target channel \( i_l \in I_n \) (with \( I_n \) denoting the set of assigned channels at timeslot \( n \)) and \( i_{\ell'} \in I_n \), respectively, for data transmission, denoted by \( a^I_n = i_l \) and \( a^{II}_n = i_{\ell'} \). When \( a^I_n = a^{II}_n \), a collision occurs and the SA perceives the collision as an indicator of severe competition for a common best channel between the two users. As a consequence, whenever a collision is observed, with probability \( p_c \), the SA will reallocate a new set of \( m \) channels with indices \( I_{n+1} = \{i'_1, ..., i'_m\} \) (selected uniformly at random) to the users for the next timeslot \( n + 1 \), hoping to stimulate readjustment in users’ channel selection strategies so as to mitigate collisions; when no collision occurs (i.e., \( a^I_n \neq a^{II}_n \)), the assigned \( m \) channels remain the same. The reward of user-I is defined in terms of throughput and is given by

\[
\begin{align*}
    r^I_n &= R^I(s_n, a^I_n, a^{II}_n) \\
    &= B \cdot \log\left\{ 1 + P \cdot \frac{h^i_{n, a^I_n}}{N} \right\} \cdot 1_{\{a^I_n \neq a^{II}_n\}},
\end{align*}
\]

where \( B \), \( P \) and \( N \) are the channel bandwidth, the transmit power and the noise power, while \( 1_{\{\cdot\}} \) is the indicator function. The reward of user-II is type-dependent and will be discussed in the next subsection. Note that the models adopted in this chapter are exemplary, and can be adjusted to fit specific contexts without influencing the main ideas presented below.

\(^2\)Among these users, malicious ones may exist, and in this chapter, we only consider insider malicious users (e.g., the compromised users) which are unknown to the SA.
6.1.2 Stochastic Game with Complete Information

Complete information SAGs, where user types are publicly known, are discussed in this subsection from user-I’s perspective. In particular, two different types, friend and enemy, are considered for user-II.3

**Enemy-type**

For an enemy-type user-II (e.g., an insider jammer), its purpose is to degrade user-I’s performance to the maximum extent rather than to gain a reward in terms of throughput. Such a malicious objective of user-II can be captured by the common zero-sum assumption on the reward functions, i.e., $R_{II}(e) = -R^I$ and $r_{II}(e) = -r^I$.4

When facing the enemy-type user-II, user-I can employ the minimax-Q algorithm [107], a widely adopted RL algorithm for zero-sum SG, to learn the best strategy in the corresponding SAG. Particularly, in minimax-Q, the (minimax) optimal quality function $Q^{(e)}(s, a^I, a^H)$ of a state-action pair $(s, a^I, a^H)$ for user-I is defined as the total expected discounted reward attained by taking action $a^I$ given state $s$ and user-II’s action $a^H$, and then following the optimal strategy from then on, and is given by5

$$Q^{(e)}(s, a^I, a^H) \triangleq R^I(s, a^I, a^H) + \mathbb{E}_{s'} \left[ \beta \cdot V^{(e)}(s') \right],$$

(6.2)

where the optimal value $V^{(e)}(s)$ of state $s$ is defined as

$$V^{(e)}(s) \triangleq \max_{\pi^I(s)} \min_{a^H} \sum_{a^I} Q^{(e)}(s, a^I, a^H) \pi^I_a(s),$$

(6.3)

with $\pi^I_a(s)$ denoting the probability that user-I takes action $a^I$ at state $s$ under strategy $\pi^I$. At each timeslot $n$, after observing the actions $a^I_n$ and $a^H_n$ of both users, the reward $r^I_n$, and the state transition from $s_n$ to $s_{n+1}$, user-I updates its learned quality and value functions by

$$Q^{(e)}_{n+1}(s, a^I, a^H) = \begin{cases} (1 - \alpha_n)Q^{(e)}_n(s, a^I, a^H) + \alpha_n \left[ r^I_n + \beta \cdot V^{(e)}(s_{n+1}) \right], & \text{for } (s, a^I, a^H) = (s_n, a^I_n, a^H_n), \\ Q^{(e)}_n(s, a^I, a^H), & \text{otherwise}, \end{cases}$$

(6.4)

$$V^{(e)}_{n+1}(s) = \max_{\pi^I(s)} \min_{a^H} \sum_{a^I} Q^{(e)}_{n+1}(s, a^I, a^H) \pi^I_a(s).$$

(6.5)

---

3In this chapter, user type is completely specified by its payoff function.
4The superscripts $(e)$ and $(f)$ are used throughout this chapter to differentiate similar quantities for the cases of enemy-type and friend-type user-II.
5In (6.2), $\mathbb{E}_{s'}$ denotes the expectation over the future state $s'$. 

101
The corresponding strategy is given by
\[
\pi^{(e)}_{n+1}(s) = \arg \max_{\pi^I(s)} \min_{a^I \in A^I} \sum_{a^I} Q^{(e)}_{n+1}(s, a^I, a^H) \pi_{a^I}^I(s). \tag{6.6}
\]

The above minimax-Q iteration ensures that \( Q^{(e)}_n (\pi^{(e)}_n) \) converges to the \( Q^{(e)}_* (\pi^{(e)}_*) \) of the SG, given that 1) all state-action pairs are visited infinitely often, and that 2) the learning rate \( \alpha_n \in (0, 1) \) admits \( \sum_{n=1}^{\infty} \alpha_n = \infty \) and \( \sum_{n=1}^{\infty} \alpha^2_n < \infty \). To satisfy the first requirement, the player has to explore (i.e., play actions uniformly at random) with a positive probability \( p_{\text{explr}} \) while exploit (i.e., follow the learned strategy) with probability \( 1 - p_{\text{explr}} \) [107]. Similarly, the enemy-type user-II can also adopt the minimax-Q to find its best strategy as user-I, by switching their roles.

**Friend-type**

When user-II is friend-type, its reward function is defined similar to that of user-I and is given by
\[
r^{H,(f)}_n (s, a^I_n, a^H_n) = R^{H,(f)}_n (s, a^I_n, a^H_n) = B \cdot \log \left( 1 + \frac{P \cdot h_n^{a^H_n}}{N} \right) \cdot 1_{\{a^H_n \neq a^I_n\}}, \tag{6.7}
\]
where \( R^{H,(f)}_n \) is assumed known to user-I. It is further assumed that when user-I knows that user-II is friend-type, both users are willing to collaborate with each other to maximize their sum rewards, but they have to learn the corresponding strategies solely based on their own local information. In this case, minimax-Q no longer conforms to the user’s objective; instead, the user can replace the max min operator in minimax-Q with a max max operator to expect and facilitate collaboration. More specifically, at each timeslot \( n \), after observing the sample \((s_n, a^I_n, a^H_n, r^I_n, s_{n+1})\), the quality and value functions of user-I are updated by
\[
Q^{(f)}_{n+1}(s, a^I, a^H) = \begin{cases} 
(1 - \alpha_n)Q^{(f)}_n(s, a^I, a^H) + \alpha_n \left[ (a^I_n + R^{H,(f)}_n(s, a^I, a^H)) 
+ \beta \cdot V^{(f)}_n(s_{n+1}) \right] & \text{for } (s, a^I, a^H) = (s_n, a^I_n, a^H_n), \\
Q^{(f)}_n(s, a^I, a^H) & \text{otherwise},
\end{cases}
\]
\[
V^{(f)}_{n+1}(s) = \max_{\pi^I(s)} \max_{a^H \in A^H} \sum_{a^I} Q^{(f)}_{n+1}(s, a^I, a^H) \pi_{a^I}^I(s). \tag{6.8}
\]

The corresponding strategy is given by
\[
\pi^{(f)}_{n+1}(s) = \arg \max_{\pi^I(s)} \max_{a^H \in A^H} \sum_{a^I} Q^{(f)}_{n+1}(s, a^I, a^H) \pi_{a^I}^I(s). \tag{6.9}
\]
The friend-type user-II will conduct a similar procedure to find its optimal collaboration strategy. Note that, when both users adopt identical learning parameters, such a collaborative learning process is equivalent to the Friend-Q learning in [128] with a common reward function $R^I + R^{II,(f)}$; when initial values are also identical, it becomes the well-known Q-learning algorithm [129] by treating the two users as a single mega-user with a composite action space $A^I \times A^{II}$. In both cases, suitable learning rate and exploration as in minimax-Q are needed for convergence assurance.

6.1.3 Spectrum Access Game with Incomplete Information

Based on the previous discussions, user-I can employ existing RL algorithms to address the complete information SAG nicely, when the type of user-II is known. However, such type information may not always be available in practice, considering that malicious users may disguise as friendly ones by using faked ID or compromising legitimate users, which will lead to incomplete information SG. Specifically, for the incomplete information SAG considered in this chapter, it is assumed that user-I is aware of the type-dependent reward functions $R^{II,(e)}$ and $R^{II,(f)}$, but it neither knows the true type of user-II nor can observe its real instant reward $r^{II}_n$; for simplicity, it is further assumed that user-II knows perfectly about user-I’s type as in the complete information SAG.

As the optimal strategies of user-I may be significantly different when facing a friend-type and an enemy-type user-II, it is crucial to conduct proper type identification in the incomplete information SAG and adjust the strategy learning process according to the identification result, so as to achieve the best performance. However, to the best of our knowledge, none of the existing RL algorithms can be applied directly to fulfill this objective.

6.2 Proposed Joint RL and Type Identification Algorithm

In this section, a joint RL and type identification algorithm that can simultaneously learn the other user’s type and the corresponding optimal responding strategy is proposed for the incomplete information SAG. In the proposed algorithm, type identification is conducted through analyzing the action history of user-II, with the rationale that action is type-driven and hence reveals type information.

In particular, user-I maintains a belief vector $b = [b^{(f)}, b^{(e)}]$ with $b^{(f)}$ and $b^{(e)}$ representing its estimates of the probabilities that user-II is a friend and an enemy, respectively, and $b^{(f)} + b^{(e)} = 1$. In addition, user-I keeps on updating two quality functions $Q^{(e)}$ and $Q^{(f)}$ according to (6.4) and (6.8) as the game evolves, so as to compute two auxiliary strategies $\pi^{(e)}$ and $\pi^{(f)}$ using (6.6) and (6.9) as the responses for enemy- and friend-type user-II, respectively. With these
auxiliary strategies, at each timeslot $n$, user-I constructs a weighted strategy

$$\pi_n^I = b_n^{(e)} \cdot \pi_n^{(e)} + b_n^{(f)} \cdot \pi_n^{(f)},$$

(6.10)

based on its current belief vector $b_n$, and then takes action according to $\pi_n^I$ (with suitable exploration).

To identify the type of user-II, user-I first constructs solely based on its local information two type-dependent strategies $\hat{\pi}_n^{II,(e)}$ and $\hat{\pi}_n^{II,(f)}$ that serve as the estimates of the real strategy taken by user-II when it is enemy-type and friend-type, respectively; then user-I adjusts its belief vector $b_n$ by comparing the observed actions $a_n^{II}$ from user-II with these two estimates. In particular, to compute $\hat{\pi}_n^{II,(e)}$, user-I can use $Q_n^{II,(e)} = -Q_n^{(e)}$ as an estimate of the quality function learned by an enemy-type user-II, considering the zero-sum assumption; and then $\hat{\pi}_n^{II,(e)}$ is updated by

$$\hat{\pi}_{n+1}^{II,(e)}(s) = \arg \max_{\pi_{II}(s)} \min_{a' \in A^I} \sum_{a''} Q_{n+1}^{II,(e)}(s,a',a'') \pi_{a''}^{II}(s).$$

(6.11)

Similarly, user-I can use $Q_n^{II,(f)} = Q_n^{(f)}$ as an estimate of the quality function learned by a friend-type user-II, and can compute the corresponding type-dependent strategy $\hat{\pi}_n^{II,(f)}$ by

$$\hat{\pi}_{n+1}^{II,(f)}(s) = \arg \max_{\pi_{II}(s)} \sum_{a'} Q_{n+1}^{II,(f)}(s,a',a'') \pi_{a''}^{II}(s).$$

(6.12)

With these two type-dependent strategies, user-I further adopts Bayes’ formula to update its belief based on user-II’s action $a_n^{II}$ as follows:

$$b_n^{(f)} = b_n \cdot L_n^{(f)} / \left( b_n^{(f)} \cdot L_n^{(f)} + b_n^{(e)} \cdot L_n^{(e)} \right),$$

(6.13)

where the likelihoods $L_n^{(f)}$ and $L_n^{(e)}$ of taking action $a_n^{II}$ at state $s_n$ given friend- and enemy-type user-II with exploration probability $p_{expl}$ are given by

$$L_n^{(f)} = (1 - p_{expl}) \cdot \pi_{a_n^{II},n}^{II,(f)}(s_n) + p_{expl} / m,$$

and

$$L_n^{(e)} = (1 - p_{expl}) \cdot \pi_{a_n^{II},n}^{II,(e)}(s_n) + p_{expl} / m,$$

(6.14)  (6.15)

respectively, with $\pi_{a_n^{II},n}^{II,(f)}(s_n)$ and $\pi_{a_n^{II},n}^{II,(e)}(s_n)$ denoting the estimated probabilities of user-II taking action $a_n^{II}$ at state $s_n$ dictated by type-dependent strategies $\hat{\pi}_n^{II,(f)}$ and $\hat{\pi}_n^{II,(e)}$, respectively; and

\*\*When user-II is truly an enemy and uses the same initial values and learning parameters as user-I, user-II’s quality function equals exactly to $Q^{II,(e)}$; similar result applies to $Q^{II,(f)}$ for friend-type user-II as well.\*\*
accordingly,
\[ b_{n+1}^{(e)} = 1 - b_{n+1}^{(f)}. \]  
\(6.16\)

The proposed joint RL and type identification scheme for the SAG with incomplete information is summarized in Algorithm 6, and its convergence property is presented in Proposition 11.

**Algorithm 6** Joint RL and Type Identification for SAG.

Initialization: \( b_0 = [0.5, 0.5], Q_0^{(e)} = 0, Q_0^{(f)} = 0, V_0^{(e)} = 0, V_0^{(f)} = 0 \) and \( \pi_0^{(e)}, \pi_0^{(f)}, \hat{\pi}_0^{II),(e), \hat{\pi}_0^{II,(f)} \) uniform.

Taking action \( a_{I,n} \) at current state \( s_n \)
- uniformly at random with probability \( p_{\text{explr}} \);
- otherwise, with probability \( \pi^{I}_{a_{I,n}}(s_n) \).

Learning: after receiving a reward \( r_{I,n} \) and observing the state transition from \( s_n \) to \( s_{n+1} \)
- Update \( Q^{(e)} \) and \( Q^{(f)} \) using (6.4) and (6.8), respectively;
- Update \( V^{(e)}, \pi^{(e)}, V^{(f)} \) and \( \pi^{(f)} \) using (6.5), (6.6), (6.8) and (6.9), respectively;
- Update \( \pi^{I} \) using (7.14);

Identification
- Update belief \( b \) using (7.16)–(6.16);
- Update \( \hat{\pi}^{II,(e)} \) and \( \hat{\pi}^{II,(f)} \) using (6.11) and (6.12), respectively;

Repeat.

**Proposition 11.** Regardless of the possible differences in the initial values and learning parameter \( \alpha \) adopted, if both users use the same discounting factor \( \beta \) and exploration probability \( p_{\text{explr}} \), the belief \( b_n \) converges to the truth, and the weighted strategy \( \pi^{I}_n \) converges to the corresponding optimal one.

**Proof.** Please see Appendix E. \( \square \)

**Remark:** Note that the requirement for identical \( \beta \) and \( p_{\text{explr}} \) is a sufficient condition (and hence may be a bit demanding in practice). Nonetheless, as shown in the next section, the

\footnote{In Algorithm 6, \( \pi^{I}_{a_{I,n}}(s_n) \) denotes the probability of user-I taking action \( a_{I,n} \) at state \( s_n \) based on its learned strategy \( \pi^{I}_n \) in (7.14).}
proposed algorithm can still be effective even when two users adopt different $\beta$'s and $p_{explr}$'s.

6.3 Numerical Results

To examine the effectiveness of the proposed method, the following SAG is considered: There are $k = 3$ channels and the corresponding channel power gains $h_1, h_2, h_3$ vary according to three different Markov processes with states $\mathcal{H}_1 = \{0, 1\}$, $\mathcal{H}_2 = \{0, 1.83\}$ and $\mathcal{H}_3 = \{0, 2\}$ and transition probability matrices $\begin{bmatrix} 0.7 & 0.3 \\ 0.2 & 0.8 \end{bmatrix}$, $\begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}$ and $\begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$, respectively. When collision occurs at the previous timeslot, with probability $p_c = 0.8$, a new set of $m = 2$ (out of $k = 3$) channels will be assigned uniformly at random to the users. The bandwidth $B$ is set to 10, and the transmit power $P$ and noise power $N$ are set to 1; for user-I, the discounting factor $\beta_I$ is set to 0.75 and the exploration probability $p_{explr} = 0.2$, and the learning rate $\alpha_n = (\mu_I)^n$ with parameter $\mu_I \in (0, 1)$ is chosen according to [107]. The average accumulative reward $\bar{r}_n$, defined as $\bar{r}_n = \frac{1}{n} \sum_{i=1}^{n} r^I_i$ [115], is the performance metric of interest for user-I. All the performance curves shown below are averaged over 1000 Monte Carlo runs.

6.3.1 Identical Initial Values and Learning Parameters

When user-II adopts the same initial values ($Q_0^H = 0, V_0^H = 0$) and learning parameters ($\beta^H = \beta^I, p_{explr}^H = p_{explr}^I, \mu^H = \mu^I$) as user-I, the type identification performance of the proposed algorithm in the SAG with incomplete information is shown in Fig. 6.2. It can be seen that the proposed method eventually achieves perfect identification in both cases of enemy-type (Fig. 6.2a) and friend-type (Fig. 6.2b) user-II. The corresponding throughput performances of user-I are compared in Fig. 6.3. As it can be seen, in both cases, the performance of the proposed method (denoted by “Bayesian”) approaches that of the ideal case (denoted by “Perfect”) where the true type of user-II is known perfectly; and it outperforms the cases of random guess (denoted by “0.5/0.5”), where user-I holds a constant belief vector $b = [0.5, 0.5]$, and wrong type (denoted by “Wrong”), where user-I makes a completely wrong type identification.

6.3.2 Different Initial Values and Learning Parameter $\alpha$

In practice, an enemy-type user-II may adopt different initial values and learning parameters as compared to user-I; while even a friend-type user-II may start the learning process at a different time (which will result in a different initial values from user-I’s viewpoint). To examine the proposed algorithm in such cases, the quality and value functions of user-II are initialized as $Q_0^H = \Delta Q$ and $V_0^H = \Delta V$ where the entries in $\Delta Q$ and $\Delta V$ are uniformly distributed in $[-\delta, \delta]$.
with $\delta = 3$; the learning parameter $\mu$ is chosen uniformly at random from $[0.8\mu, 1.3\mu]$. The corresponding type identification and throughput performance are presented in Fig. 6.4 and Fig. 6.5, respectively. It can be seen from Fig. 6.4 and Fig. 6.5 that the proposed method can still effectively identify the true type and the corresponding throughput performance approaches that of the ideal case gradually, which conforms to Proposition 11. Besides, by comparing Fig. 6.2 with Fig. 6.4, it is worth noting that a longer learning time is required by the proposed algorithm when user-II uses different initial values and learning parameter $\alpha$.

### 6.3.3 Different $\beta$ and $p_{ex}l$r

When user-II further uses different discounting factor $\beta$ and exploration probability $p_{ex}l$r chosen randomly from $[0.5, 0.9]$ and $[0.25p_{ex}l, 1.5p_{ex}l]$, respectively, the corresponding type identification performance is presented in Fig. 6.6. It can be seen that the proposed algorithm still successfully detects the true type (with nearly identical learning speed as compared to the

---

8The average value of the entry in the quality function is about 2.5 in the considered scenario. Therefore, we consider $\delta = 3$ as a sufficient disturbance.
case of equal $\beta$’s and $p_{explr}$’s shown in Fig. 6.4); the corresponding throughput performance is similar to Fig. 6.5 and is omitted in the interest of space. The intuition is that in the considered scenario, the discounting factor does not have a very significant impact on the learned strategy while the difference in exploration probability does not obfuscate user-II’s type-dependent strategies much. In such cases, even though the requirement on $\beta$ and $p_{explr}$ in Proposition 11 is not satisfied, the proposed algorithm can still be quite effective.

6.4 Related Works

As a promising mathematical tool for modeling and analyzing multi-party decision making processes, game theory has been widely applied to wireless communications and related security problems ([30, 130] and references therein). Nevertheless, the issues of imperfect monitoring and incomplete information often exist when game theoretic models are applied in practice; the former refers to that other players’ actions are not always (perfectly) observable, while the
latter concerns about uncertainties about the opponent such as its type and is the focus of this chapter.

In existing works (e.g., [34, 35]) on incomplete information (classical) games, the player’s optimal strategy is often derived with respect to a prior distribution on other players’ types through a Bayesian approach. However, since the prior does not necessarily always match the truth, such approaches often lead to a performance degradation as compared to the corresponding games with complete information [116]. When the game is played repeatedly, the player can update its belief on the type of the other player using past observations [36, 37]. In contrast, this chapter considers incomplete information in the context of SG, which is more suitable for unknown dynamic wireless environments in practice.

As compared to incomplete information games in classical game theory, incomplete information SG is relatively less explored in literature. When the incomplete information is described by a finite collection of SGs and the state transition only depends on one player’s action, [131] studies the existence of the value function of the corresponding SG. In [132], a two-player nonzero-sum SGs with incomplete information is formulated to model the competitions in a network security game; but the corresponding state transitions are independent of the player actions and the current state, which is a special case of the standard SG modeling. In [133], the unknown state transition statistics are considered as the source of incomplete information and the corresponding optimal strategy is analyzed under the assumption that player always assumes worst-case state transitions. Different from these pioneering works on incomplete information SG, our work in this chapter adopts the standard SG model and considers incomplete information in terms of the other player’s type.
6.5 Conclusions

In this chapter, a joint RL and type identification algorithm is proposed for SG with incomplete information. Built upon the conventional RL algorithm, the proposed algorithm enables a player to gradually update its belief on the type of the other player. Under certain (sufficient) conditions, the proposed algorithm provably ensures that the belief and the learned strategy converge to the truth and the optimal one, respectively. Using the SAG as an example, numerical results confirm the effectiveness of the proposed algorithm. Moreover, it is observed that even when some required conditions for convergence are not satisfied, the proposed algorithm can still be effective in the SAG.
Chapter 7

Dynamic IDS Configuration in the Presence of Intruder Type Uncertainty

This chapter further extends the zero-sum incomplete information SG studied in the previous chapter to the more general non-zero-sum incomplete information SG, for the application of intrusion detection system (IDS) configuration.

Information systems and networks are vulnerable to various forms of intrusions, leaving their security an ever-present concern. As an effective approach to providing security protection to information systems through monitoring the system/network usage for malicious activities and policy violations, IDSs have assumed paramount importance in this information era, and thus have promoted extensive research efforts in the past decades (see, e.g., [134–136] and the references therein).

Although various intrusion detection algorithms have been developed in literature, the attacks from intruders are also diversified. It is very unlikely that any single intrusion detection algorithm can catch all possible kinds of intrusions. In addition, practical IDS usually has only limited resource and simultaneously running all the existing algorithms may not be feasible for real-time intrusion detection [137]. For these reasons, it is crucial to (dynamically) configure the IDS with a proper library of intrusion detection algorithms that can effectively detect the attacks from the intruder without exceeding the resource limitation. To this end, when the objective, or equivalently the type, of the intruder is known, game theory can be employed to analyze and predict the intruder’s behaviors based on its objective so as to facilitate IDS configurations [138, 139]. But in practice, an IDS may encounter different intruders whose types are seldom known beforehand - sometimes the attacks may be merely due to misbehaviors of a negligent legitimate user; such intruder type uncertainty complicates the IDS configuration
problem. Another difficulty for IDS configuration is that variations in resource availability and work load and influence from the intrusions and corresponding defenses may induce (often unknown) system state dynamics.

Considering these two issues, the incomplete information stochastic game (SG) seems a natural model for the above IDS configuration problem. Specifically, SG with complete information considers the interactions among intelligent players in a dynamic system, and when the type of the other player is unknown, it becomes an incomplete information SG [33, 116]. For SG with complete information, when the system dynamism is unknown, a player can employ reinforcement learning algorithms (e.g., Nash Q-learning [140]) to gradually learn its strategy through interactions with both the system and the other player. To address the incomplete information SG, a new algorithm, termed Bayesian Nash-Q learning, that combines the conventional Nash-Q learning with a Bayesian type identification procedure is proposed in this chapter. This chapter mainly focuses on the application of IDS configuration, but the proposed algorithm is general and may assume wider applications.

The remainder of this chapter is organized as follows. Section 7.1 formulates the IDS configuration problem and Section 7.2 introduces the background of SG and the Nash-Q learning algorithm. The proposed Bayesian Nash-Q learning algorithm is presented in Section 7.3 and its effectiveness is examined numerically in Section 7.4. Related works are discussed in Section 7.5. Conclusions are presented in Section 7.6.

### 7.1 Problem Formulation

In this chapter, the following IDS configuration problem is considered. Particularly, it is assumed that the intruder can launch $N$ different attacks $A = \{a_1, ..., a_N\}$ and the IDS has $N$ libraries $L = \{l_1, ..., l_N\}$ of intrusion detection algorithms; and library $l_i$ can detect attack $a_i$ with a high probability $p_{i,i}$ but other attacks $\{a_j\} (j \neq i)$ with low probabilities $\{p_{i,j}\}$. Without loss of generality, it is also assumed that the IDS can only load one library at a time due to resource limitation. To model the influence of library configuration $l$ and attack $a$ on the state of the target system protected by the IDS, it is assumed that the target system state will transit from $s$ to $s'$ with certain (unknown) probability $p_{st}(s'|s, l, a)$. When the target system is at state $s$ and the IDS loads library $l_i$ while the intruder takes attack $a_j$, the reward function of the IDS is modeled by

$$R^{IDS}(s, l_i, a_j) = p_{i,j} w^{IDS}_{s,j}. \quad (7.1)$$

Some interpretations for this reward model are in order. 1) The modeling in (7.1) conforms to the intuition that the IDS can obtain good reward only when it uses the right library that can detect the attack with high probability. 2) Considering that the damages caused by the
same attack may be different at different system states, the factor $w_{s,j}^{IDS}$ is used to reflect the importance of detecting attack $a^j$ at a given state $s$.

Moreover, it is further assumed that the intruders of interest belong to a known set, denoted by $\Gamma$. The reward function of a type-$\gamma$ ($\gamma \in \Gamma$) intruder is modeled as

$$R^{\gamma}(s, l^i, a^j) = (1 - p_{i,j})w_{s,j}^{\gamma},$$

(7.2)

where $w_{s,j}^{\gamma}$ denotes the profit of fulfilling attack $a^j$ at state $s$ for type-$\gamma$ intruder. Due to the difference in rewards, the IDS should expect different attacking strategies for different types of intruders; further considering the resource limitation, the IDS has to strategically select its detection libraries that match well to the potential attacks from the intruder based on its type. However, in practice, the real objective and type of the intruder can rarely be known beforehand, though the set of payoff functions $\{R^{\gamma}\}_{\gamma \in \Gamma}$ may be given (e.g., by security analysts). Therefore, the IDS has to conduct intruder type identification jointly with intrusion detection library configuration.

### 7.2 Stochastic Game and Nash-Q

In this section, we consider a simplified IDS configuration problem by assuming that the intruder type is known. To this problem, we first introduce the basics of SG and then demonstrate how to use Nash-Q algorithm [140] to solve it.

#### 7.2.1 Stochastic Game

When the intruder’s type $\gamma$ is known, the IDS configuration problem described in Section 7.1 can be formulated as a (complete information) SG [33] as follows: the IDS and the intruder are the two players and the set of possible target system states $\mathcal{S}$ defines the state space of this SG; the set of available libraries $\mathcal{L}$ and the set of possible attacks $\mathcal{A}$ correspond to the action spaces for the IDS and the intruder, respectively; $R^{IDS}$ and $R^\gamma$ define the reward functions for these two players, respectively; and $p_{st}(s'|s, l, a)$ is the conditional state transition function that specifies the probability of reaching a future state $s' \in \mathcal{S}$, given the current state $s \in \mathcal{S}$ and action pair $(l, a) \in \mathcal{L} \times \mathcal{A}$. The objective of the IDS (intruder) 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 $\beta \in [0, 1)$, which represents its long-term performance with diminishing weighting on the future. To this end, the IDS (intruder) needs to learn a strategy $\pi_n^{IDS}(s, l)$ ($\pi_n^{\gamma}(s, a)$) that specifies the probability of taking action $l \in \mathcal{L}$ ($a \in \mathcal{A}$) at a given state $s$ through proper foresighted planning. At each timeslot $n$, both the IDS and the intruder observe the current state $s_n \in \mathcal{S}$ and take actions $l_n \in \mathcal{L}$ and $a_n \in \mathcal{A}$ according to their learned strategies $\pi_n^{IDS}$ and $\pi_n^{\gamma}$, respectively; then, they receive
instant rewards $r_n^{IDS} = R^{IDS}(s_n, l_n, a_n)$ and $r_n^{γ} = R^{γ}(s_n, l_n, a_n)$, respectively. Meanwhile, the target system transits to a new state $s_{n+1} \in S$ according to the (often unknown) state transition function $p_{st}$.

### 7.2.2 Nash-Q

The Nash-Q learning algorithm [140] can be employed to solve the SG described in Section 7.2.1. In the Nash-Q learning, the optimal quality function $Q^{IDS,γ}_*(s, l, a)$ of a state-action pair $(s, l, a)$ for the IDS (against a type-$γ$ intruder), which represents the total expected discounted reward attained by taking action $l$ given the state $s$ and intruder’s action $a$ and then following the optimal strategy from then on, is defined as

$$Q^{IDS,γ}_*(s, l, a) \triangleq R^{IDS}(s, l, a) + \mathbb{E}_{s'} \left[ \beta \cdot V^{IDS,γ}_*(s') \right],$$  

(7.3)

and similarly, the optimal quality function $Q^γ_*(s, l, a)$ for the type-$γ$ intruder is defined as

$$Q^γ_*(s, l, a) \triangleq R^γ(s, l, a) + \mathbb{E}_{s'} \left[ \beta \cdot V^γ_*(s') \right],$$  

(7.4)

where for any state $s$, the optimal value functions $V^{IDS,γ}_*$ and $V^γ_*$ for the IDS and the intruder, respectively, are defined as

$$V^{IDS}_*(s) \triangleq \text{NASH}^{IDS}(Q^{IDS,γ}_*(s, \cdot, \cdot), Q^γ_*(s, \cdot, \cdot)), \quad (7.5)$$

and

$$V^γ_*(s) \triangleq \text{NASH}^γ(Q^{IDS,γ}_*(s, \cdot, \cdot), Q^γ_*(s, \cdot, \cdot)). \quad (7.6)$$

Note that in (7.5) and (7.6), $\text{NASH}^{IDS}(Q^{IDS,γ}_*(s, \cdot, \cdot), Q^γ_*(s, \cdot, \cdot))$ represents the reward of the IDS at a Nash equilibrium (NE) of an equivalent single stage game with corresponding reward functions of the two players specified by the two matrices $Q^{IDS,γ}_*(s, \cdot, \cdot)$ and $Q^γ_*(s, \cdot, \cdot)$, respectively, and $\text{NASH}^γ(Q^{IDS,γ}_*(s, \cdot, \cdot), Q^γ_*(s, \cdot, \cdot))$ is defined similarly as the reward of the type-$γ$ intruder. For each state $s$, the NE strategies for the IDS and the intruder in the SG are the NE strategies that generate rewards $V^{IDS}_*(s)$ and $V^γ_*(s)$, respectively, in the corresponding equivalent single stage game. For ease of presentation, they are written respectively as

$$\pi^{IDS,γ}_*(s, \cdot) = \text{arg} \text{NASH}^{IDS}(Q^{IDS,γ}_*(s, \cdot, \cdot), Q^γ_*(s, \cdot, \cdot)).$$  

(7.7)

$^1$In (7.3), $\mathbb{E}_{s'}$ denotes the expectation over the future state $s'$. 

114
and

\[ \pi_n^\gamma(s, \cdot) = \text{arg} \text{NASH}^\gamma(Q^{\text{IDS}, \gamma}_{n+1}(s, \cdot, \cdot), \tilde{Q}^\gamma_{n+1}(s, \cdot, \cdot)). \quad (7.8) \]

It can be noticed from (7.7) and (7.8) that, to obtain the NE strategies, both players need not only its own quality function but also that of the opponent. Therefore, at each timeslot \( n \), after observing the actions \((l_n, a_n)\), the reward \( r^{\text{IDS}}_n \), and the state transition from \( s_n \) to \( s_{n+1} \), the IDS updates the quality and the value functions of its own and further maintains a pair of virtual quality and value functions, \( \tilde{Q}^\gamma \) and \( \tilde{V}^\gamma \), for the type-\( \gamma \) intruder to keep track of its behaviors, as follows:

\[
Q^{\text{IDS}, \gamma}_{n+1}(s, l, a) = \begin{cases} 
(1 - \alpha_n) Q^{\text{IDS}, \gamma}_n(s, l, a) + \alpha_n \left[ r^{\text{IDS}}_n + \beta \cdot \tilde{V}^\gamma_n(s_{n+1}) \right] 
, & \text{for } (s, l, a) = (s_n, l_n, a_n), \\
Q^{\text{IDS}, \gamma}_n(s, l, a), & \text{otherwise}, 
\end{cases} \quad (7.9)
\]

\[
\tilde{Q}^\gamma_{n+1}(s, l, a) = \begin{cases} 
(1 - \alpha_n) \tilde{Q}^\gamma_n(s, l, a) + \alpha_n \left[ R^\gamma(s, l, a) + \beta \cdot \tilde{V}^\gamma_n(s_{n+1}) \right] 
, & \text{for } (s, l, a) = (s_n, l_n, a_n), \\
\tilde{Q}^\gamma_n(s, l, a), & \text{otherwise}, 
\end{cases} \quad (7.10)
\]

\[
V^{\text{IDS}}_{n+1}(s) = \text{NASH}^{\text{IDS}}(Q^{\text{IDS}, \gamma}_{n+1}(s, \cdot, \cdot), \tilde{Q}^\gamma_{n+1}(s, \cdot, \cdot)). \quad (7.11)
\]

\[
\tilde{V}^\gamma_{n+1}(s) = \text{NASH}^\gamma(Q^{\text{IDS}, \gamma}_{n+1}(s, \cdot, \cdot), \tilde{Q}^\gamma_{n+1}(s, \cdot, \cdot)). \quad (7.12)
\]

Then, the updated strategy of the IDS is given by

\[
\pi^{\text{IDS}, \gamma}_{n+1}(s, \cdot) = \text{arg} \text{NASH}^{\text{IDS}}(Q^{\text{IDS}, \gamma}_{n+1}(s, \cdot, \cdot), \tilde{Q}^\gamma_{n+1}(s, \cdot, \cdot)). \quad (7.13)
\]

By switching the roles, the intruder can conduct a similar learning procedure to update its strategy \( \pi_n^\gamma \) using the above equations.\(^2\) It has been shown in [140] that, with suitable learning rate \( \alpha_n \)’s, the learned quantities in the Nash-Q algorithm converge to the corresponding optimal ones under certain (restrictive) sufficient conditions.

**Issue of imperfect monitoring:** In the above Nash-Q algorithm, it is assumed that the player can always observe the opponent’s action. This assumption may not hold when the IDS fails to detect the intrusion, and in such cases, the IDS faces a SG with imperfect monitoring. To our best knowledge, solving SG with both imperfect monitoring and incomplete information is still

\(^2\)The intruder may suffer performance loss if it deviates from the NE unilaterally.
an open problem and there is no well-established solution so far. We do not attempt to solve it in this chapter. Rather, some reasonable assumptions as in the existing literature (e.g., [139]) will be taken. In particular, it is assumed in this chapter that any undetected intrusion will cause some immediate damage to the target system and the IDS can infer the intruder’s action based on the damage. In addition, when such assumptions do not hold, random conjecture on the intruder’s action could be a (naive) solution, and the corresponding numerical results will be presented in Section 7.4.2.

### 7.3 The Proposed Method

In practice, the type of the intruder is usually unknown and in such cases, the IDS needs to simultaneously identify the intruder’s type and learn the corresponding library configuration.

In the proposed Bayesian Nash-Q learning, type identification is conducted through analyzing the actions of the intruder, with the rationale that action is type-driven and hence reveals type information. In particular, the IDS maintains a belief vector $b = [b^{\gamma}]_{\gamma \in \Gamma}$ with $b^{\gamma}$ representing an estimate of the probability that the intruder’s type is $\gamma$, and $\sum_{\gamma \in \Gamma} b^{\gamma} = 1$. It is worth noting that although the real instant reward $r^{\gamma}_n$ of the intruder cannot be observed, the IDS can compute the set of type-dependent rewards $\{r^{\gamma}_n\}_{\gamma \in \Gamma}$ based on the intruder’s action. With this information, for each possible intruder type $\gamma$, the IDS keeps on updating a pair of quality and value functions, $Q^{IDS,\gamma}$ and $V^{IDS,\gamma}$, for itself as in (7.9) and (7.11), and also another pair of virtual quality and value functions, $\tilde{Q}^{\gamma}$ and $\tilde{V}^{\gamma}$, to mimic the learning process of a type-$\gamma$ intruder as in (7.10) and (7.12). Further based on these quality and value functions, the IDS computes a set of auxiliary strategies $\pi^{IDS,\gamma}$ using (7.13) as optimal responses to different types of intruders. With these auxiliary strategies, at each timeslot $n$, the IDS constructs a weighted strategy

$$\pi^{IDS}_n = \sum_{\gamma \in \Gamma} b^{\gamma}_n \cdot \pi^{IDS,\gamma}_n,$$

(7.14)

based on its current belief vector $b_n = [b^{\gamma}_n]_{\gamma \in \Gamma}$, and then takes action according to $\pi^{IDS}_n$ (with suitable exploration probability $p_{expl}$ [33]).

To identify the type of the intruder, the IDS first constructs a mimicked type-dependent strategy $\tilde{\pi}^{\gamma}_n$ for each type of intruder, which is given by

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

(7.15)

Note that the computation of $\tilde{\pi}^{\gamma}_n$ only requires local information maintained by the IDS and it serves as an estimate of the real strategy $\pi^\gamma_n$ taken by the type-$\gamma$ intruder. The IDS can
compare the observed (or inferred) action of the intruder with these mimicked strategies so as to adjust the belief vector $b_n$ for intruder type identification. Specifically, the IDS adopts the Bayes’ formula to update its belief based on the intruder’s action $a_n$ as follows:

$$b_{n+1}^\gamma = b_n^\gamma \cdot L_n^\gamma / \sum_{\gamma' \in \Gamma} b_n^{\gamma'} \cdot L_n^{\gamma'} , \quad \forall \gamma \in \Gamma,$$

(7.16)

where the likelihood $L_n^{\gamma'}$ of the type-$\gamma'$ intruder taking action $a_n$ at state $s_n$ is given by

$$L_n^{\gamma'} = (1 - p_{explr}) \cdot \tilde{\pi}_n^{\gamma'}(s_n, a_n) + p_{explr}/N, \quad \forall \gamma' \in \Gamma$$

(7.17)

with $\tilde{\pi}_n^{\gamma'}(s_n, a_n)$ denoting the estimated probability of the intruder taking action $a_n$ at state $s_n$ dictated by the type-dependent strategy $\tilde{\pi}_n^{\gamma'}$.

The proposed method is summarized in Algorithm 7.

**Algorithm 7** Bayesian Nash-Q learning for IDS configuration

Initialization: $Q_0^{IDS,\gamma} = 0$, $\tilde{Q}_0^\gamma = 0$, $V_0^{IDS,\gamma} = 0$, $\tilde{V}_0^\gamma = 0$ and $\tilde{\pi}_0^{IDS,\gamma}$, $\tilde{\pi}_0^\gamma$, $b$ are uniformly distributed.

Taking action $l_n$ at current state $s_n$

- uniformly at random with probability $p_{explr}$;
- otherwise, with probability $\pi_n^{IDS}(s_n, l_n)$.

Learning: after receiving a reward $r_n^{IDS}$ and observing the system state transition from $s_n$ to $s_{n+1}$

- Update $Q_n^{IDS,\gamma}$ and $\tilde{Q}_n^\gamma$ using (7.9) and (7.10), respectively, for all $\gamma \in \Gamma$;
- Update $V_n^{IDS,\gamma}$, $\pi_n^{IDS,\gamma}$, $\tilde{V}_n^\gamma$ and $\tilde{\pi}_n^\gamma$ using (7.11), (7.13), (7.12) and (7.15), respectively, for all $\gamma \in \Gamma$;
- Update $\pi_n^{IDS}$ using (7.14);

Intruder type identification

- Update the belief vector $b$ using (7.16);

Repeat.

---

3In Algorithm 7, $\pi_n^{IDS}(s_n, l_n)$ denotes the probability of the IDS taking action $l_n$ at state $s_n$ based on its learned strategy $\pi_n^{IDS}$ in (7.14).
7.4 Numerical Results

Numerical results are presented in this section to evaluate the effectiveness of the proposed method. Specifically, the following scenario is considered: The IDS is interested in preventing three types of intruders $\gamma_1$, $\gamma_2$ and $\gamma_3$; all these intruders can launch three different attacks $a^1$, $a^2$ and $a^3$, and the corresponding intrusion detection libraries of the IDS are $l^1$, $l^2$ and $l^3$; the detection probability $p_{i,j}$ of library $l^i$ against attack $a^j$ is 0.85 when $i = j$, and 0.3 otherwise. The system can be in either one of the two states $s^1$ and $s^2$, and the corresponding action-dependent state transition matrices are set as $p_{st}(\cdot|\cdot, l^p, a^q) = \begin{bmatrix} 0.9 & 0.1 \\ 0.8 & 0.2 \end{bmatrix}$ for $p = q$, and $p_{st}(\cdot|\cdot, l^p, a^q) = \begin{bmatrix} 0.3 & 0.7 \\ 0.2 & 0.8 \end{bmatrix}$ for $p \neq q$, with the number at the $i$-th row and the $j$-th column representing the probability of transiting from state $s^i$ to state $s^j$. The importance factors of detecting different attacks are set as $w^{IDS} = \begin{bmatrix} 1 & 6 & 7 \\ 4 & 1 & 1 \end{bmatrix}$ with the number at the $i$-th row and the $j$-th column representing the importance of detecting attack $a^j$ at state $s^i$. Similarly, the profit factors for different types of intruders are set as $w^{r1} = \begin{bmatrix} 2 & 2 & 10 \\ 4 & 3 & 15 \end{bmatrix}$, $w^{r2} = \begin{bmatrix} 8 & 3 & 1 \\ 10 & 2 & 3 \end{bmatrix}$ and $w^{r3} = \begin{bmatrix} 2 & 9 & 1 \\ 1 & 7 & 3 \end{bmatrix}$, respectively, reflecting the consideration that different types of intruders may have different preference over these attacks. For the IDS, the discounting factor $\beta^{IDS}$ is set to 0.25 and the exploration probability $\mu^{IDS}_{expl} = 0.2$, and the learning rate $\alpha_n = (\mu^{IDS})^n$ with parameter $\mu^{IDS} \in (0, 1)$ is chosen according to [140]. While for the intruder, considering that it may adopt different initialization and learning parameters as compared to the IDS, its initial quality and value functions are set to $Q^0_0 = \Delta Q$ and $V^0_0 = \Delta V$ where the entries of $\Delta Q$ and $\Delta V$ are uniformly distributed in the range $[-\delta, \delta]$ with $\delta = 2.4$. Also, the learning parameter $\mu^\gamma$, the exploration probability $\mu^{expl\gamma}$ and the discounting factor $\beta^\gamma$ of the intruder are chosen uniformly at random from $[0.8\mu^{IDS}, 1.3\mu^{IDS}]$, $[0.15, 0.25]$ and $[0.1, 0.3]$, respectively.

The average accumulative reward $\bar{r}^{IDS}_n$, defined as [115]

$$\bar{r}^{IDS}_n = \frac{1}{n} \sum_{i=1}^{n} r^{IDS}_i,$$

is the performance metric of interest for the IDS. In all simulations shown below, it is assumed that the true type of the intruder is $\gamma_1$, and all the performance curves shown below are averaged over 1000 Monte Carlo runs.

\[\text{The average value of the entries in the quality function is about 1.5 in the considered scenario. Therefore, we consider } \delta = 2 \text{ as a sufficient disturbance.}\]
7.4.1 Perfect Monitoring

The perfect monitoring case is examined first, where it is assumed that the IDS can always directly observe or indirectly infer the intruder’s actions. Fig. 7.1 shows the IDS’s belief $b_{\gamma_1}$ on the intruder’s type over time, and it can be seen that the proposed algorithm can detect the intruder’s type with fairly high fidelity.\(^5\) As a result of this accurate type identification, the IDS can properly configure the most suitable detection libraries against the intruder, and this is shown in Fig. 7.2. Particularly, the average accumulated reward of using the proposed algorithm (denoted by “Bayesian”) is compared with those of four other cases: 1) the ideal case that the IDS knows the intruder’s type beforehand and adopts the right library accordingly (denoted by “Perfect”), 2) the case that the IDS uniformly randomly guesses the intruder’s type (denoted by “Random”), and 3) the cases that the IDS makes completely wrong type identifications (denoted by “Type-2” and “Type-3”, respectively). It can be seen that the performance of the proposed algorithm is close to that of the ideal case, and it substantially outperforms the other three cases.\(^6\)

7.4.2 Imperfect Monitoring

If the IDS cannot perfectly infer the intruder’s action based on the corresponding damage, imperfect monitoring occurs. Finding optimal solutions for such problems is beyond the scope of this chapter, instead, we take a simple solution as follows: At each round, if the IDS fails to directly observe the intruder’s true action $a_n$ (with probability $1 - 0.85$ if the IDS uses the right library and $1 - 0.3$ otherwise, in the considered scenario), it will make a conjecture $a'_n \in A$.

---

\(^5\)The precision of type-identification using the proposed method does not reach 100%, since, 1) except for some very restrictive cases, the convergence of the original Nash-Q is not guaranteed [140], and so is the proposed algorithm, and 2) the initialization and the learning parameters of the intruder and the IDS are different in the considered scenario. Consequently, in some Monte Carlo runs, the mimicked strategy by the IDS may not match the true strategy of the intruder well.

\(^6\)When the true intruder type is $\gamma_2$ or $\gamma_3$, similar observations can be made, the corresponding results of which are not presented in the interest of space.
uniformly at random, and then conduct the learning procedure (7.9)–(7.16) with $a_n$ replaced by $a'_n$. The corresponding type identification performance and average accumulate reward of the IDS are presented in Fig. 7.3 and Fig. 7.4, respectively. It can be seen from Fig. 7.3 that, even in the presence of imperfect monitoring, the IDS can still maintain satisfactory type identification performance in the considered scenario. Intuitively, this is because that the effect of the random conjecture on type-identification is similar to that of a larger exploration probability. But as a suboptimal solution, it can be seen from Fig. 7.4 that imperfect monitoring does degrade the corresponding average accumulated rewards of the IDS as compared to the perfect monitoring cases shown in Fig. 7.2, even when the intruder’s type is known perfectly. Nonetheless, the IDS using the proposed algorithm still substantially outperforms those of using random or wrong intruder type, justifying the advantage of type-identification using the proposed algorithm even in the presence of imperfect monitoring.
Game theory provides a set of mathematical frameworks to compute and analyze the optimal response against intelligent and rational opponents. In the past decade, various game theoretic models have been applied to the intrusion detection problem, assisting with intruder behavior prediction and detection performance enhancement.

Early works in this direction often model the interaction between the IDS and the intruder as a single stage game. By assuming full knowledge of the game, these studies mainly focus on determining the optimal IDS actions and predicting the behaviors of the intruder based on the NE. For example, in [141], a two-player general-sum game is formulated to facilitate the IDS to choose the right subsystem for intrusion detection based on the corresponding NE. In [138], the NE strategy is used as a prediction of the intruder’s behaviors. In addition, considering that in many cases, the intruder may launch attacks based on the strategy of the IDS, the Stackelberg game (a.k.a. leader-follower game) is employed in [142] to model the corresponding security competitions. Moreover, instead of considering NE that is suitable for players with full rationality, the logit equilibrium is adopted in [143] to analyze the rivalry between the IDS and the intruder with bounded rationality. Besides these non-cooperative games, cooperative game models have also been considered. For example, coalition game is considered in [144] to guide the collaborations among a set of distributed IDS’s in mobile ad hoc networks, and the Shapley value [145] is used to express the contribution of each node in intrusion detection.

In practice, the IDS may lack knowledge about the opponent’s reward function and hence cannot directly compute the NE strategies. To address this, repeated game model has been employed in existing literature, in which both the IDS and the intruder can gradually learn the NE strategy through repeated interactions. For instance, the fictitious play learning algorithm has been employed in [146] to find the optimal strategies in a repeated intrusion detection game. In addition, in many cases, the IDS does not even know the type of its opponent (e.g., negligent normal user or real intruder [36]). In such cases, the Bayesian game, in which the
IDS keeps on updating a belief on the opponent’s type based on the opponent’s actions, can be employed [147].

Further taking target system dynamics into consideration, SG model has been employed in [139] to study the IDS configuration problem, but in [139], the type of the opponent is assumed known. Unlike in the repeated game case, the incomplete information issue of IDS in the SG setting is relatively less explored in literature. Some pioneer efforts in this direction include [132] and [133]. In [132], a two-player nonzero-sum SG with incomplete information is formulated to model the competitions in a network security game; but the corresponding state transitions are independent of the player actions and the current state, which is a special case of the standard SG modeling (and is essentially a repeated game with stochastic rewards). In [133], the unknown state transition statistics are considered as the source of incomplete information and the corresponding optimal strategy is analyzed under the assumption that players always assume worst-case state transitions. Different from these works, our work in this chapter adopts the standard SG model and considers incomplete information in terms of the other player’s type. As compared to our work presented in the previous chapter that considers a zero-sum incomplete information SG with application in wireless communications, this work presented in this chapter eliminates the zero-sum restriction and considers the more general non-zero sum incomplete information SG with application in IDS configuration.

7.6 Conclusions

In this chapter, the IDS configuration problem is modeled as an incomplete information SG, and a new algorithm, termed Bayesian Nash-Q that combines the conventional Nash-Q algorithm with a Bayesian type identification procedure is proposed to solve this incomplete information SG. Numerical results show that the proposed algorithm provides sufficiently accurate intruder type identification and hence can guide the IDS configuring its detection libraries effectively. Even in the case of imperfect monitoring, the proposed algorithm can still delivery reasonably good performance.
Chapter 8

On the Dependability of Link Signature

Besides developing new algorithms as discussed in previous chapters, another substantial aspect in the security field is to obtain comprehensive evaluation of newly developed security techniques so as to avoid false sense of security. Among the various recently advocated physical layer security techniques (e.g., [148–152]), the dependability of the recently advocated link-signature (LS) will be investigated in this chapter.

While conventional computational-complexity based cryptography has received great success, there is a haunting concern that building security on the hardness of computing problems is not worry-free. Motivated by information-theoretic security, a recent progress in practical security protocol development ripples around, where a central concept is the so-called LS. Particularly, LS refers to the ample channel characteristics between two wireless devices, which can serve as the source of common randomness for security establishment. While bringing great conveniences to security establishment, LS is broadcast in nature and hence may suffer potential vulnerabilities as any other wireless applications. Recently, several potential attacks that can severely impair the security established by LS based mechanisms have been noticed by researchers (e.g., [55, 56]). To defend against such attacks, guard zones with suitable sizes must be deployed around the legitimate devices. As to this, existing LS based security schemes often assume that the legitimate and the adversary channels are essentially uncorrelated and hence the attacker can barely acquire any useful information about the legitimate LS, as long as the adversary receiver is separated from the legitimate one by more than half a wavelength. However, high channel correlation has been observed in practice as well, even when the spatial separation is more than half-wavelength.

To resolve these seemingly contradictory facts, existing wireless channel models are revisited and several key factors that have significant influence on LS security have been identified.
Based on the obtained understandings, a generic channel correlation model is developed for LS security assessment. With this model, suitable guard zone sizes are numerically explored for LS based security mechanisms in several typical indoor and outdoor communication scenarios. Moreover, real-world measurement campaigns based on Universal Software Radio Peripheral (USRP) platforms and GNURadio are conducted in this work to further justify the analysis.

The rest of this chapter is organized as follows. Section 8.1 demonstrates the existence of high channel correlation and the potential vulnerability of LS. Important factors and models influencing channel correlations and LS security are explored in Section 8.2. Numerical and experimental results are presented in Section 8.3 and Section 8.4, respectively. Section 8.5 concludes this chapter.

8.1 Potential Vulnerability of LS and the Correlation Attack

As all the security established by LS based mechanisms rely on the confidentiality of the channel information between the corresponding legitimate transmitter \( (T_l) \) and receiver \( (R_l) \), a widely adopted assumption is that, when the legitimate receiver and the adversary receiver \( (R_a) \) are separated by more than half a wavelength \( (\lambda/2) \), the corresponding wireless channels \( h_{T_l,R_l} \) and \( h_{T_l,R_a} \) are essentially decorrelated such that the adversary can barely acquire any information about the legitimate channel [45]. Specifically, the correlation between the legitimate and the adversary channels is defined as\(^1\),

\[
\rho \triangleq \frac{\mathbb{E}[h_{T_l,R_l}h_{T_l,R_a}^*] - \mathbb{E}[h_{T_l,R_l}^*]\mathbb{E}[h_{T_l,R_a}]}{\sqrt{\text{Var}(h_{T_l,R_l})\text{Var}(h_{T_l,R_a}^*)}}. \tag{8.1}
\]

In this section, the famous one-ring model is introduced to illustrate that the half-wavelength decorrelation assumption may not be true in some scenarios and two wireless channels can be highly correlated over a much larger spatial range, after which, the correlation attack is presented to manifest the potential vulnerability of the LS in such cases.

8.1.1 Channel Correlation based on One-ring Model

Well-supported by real-world evidences [153], the one-ring model [59, 154, 155] is suitable to characterize the correlation between two wireless channels when one communication end is surrounded by rich scatterers while the other end experiences much less diffusion (Fig. 8.1). According to this model, the correlation between a pair of channels \( h_{pq} \) and \( h_{p'q'} \) is given by [155]

\(^1\)In (8.1), *, \( \mathbb{E}[\cdot] \) and \( \text{Var}(\cdot) \) denote the conjugate, the expectation and the variance operators, respectively.
Figure 8.1: One-ring model with receiver side scatterers ($S_\theta$: scatterer at azimuth $\theta$; $p, p'$: transmitters; $q, q'$: receivers).

Figure 8.2: Scatterer-ring on receiver side ($T_l$: legitimate transmitter; $R_l$: legitimate receiver; $R_a$: adversary receiver; $\delta d$: spatial separation between the legitimate and the adversary receivers).

\[
\rho_{pq,p'q'} = \int_{-\pi}^{\pi} \exp \left\{ \frac{2\pi j}{\lambda} [d_{pp'} \cos(\theta_T - \varphi) + d_{qq'} \cos(\theta_R - \theta)] \right\} f(\theta) d\theta, \tag{8.2}
\]

where $\varphi$ admits
\[
\sin(\varphi) = \Delta \sin(\theta) / \sqrt{1 + \Delta^2 + 2\Delta \cos(\theta)}, \tag{8.3}
\]
\[
\cos(\varphi) = (1 + \Delta \cos(\theta)) / \sqrt{1 + \Delta^2 + 2\Delta \cos(\theta)}. \tag{8.4}
\]

From (8.2) it can be noted that, in general, the channel correlation $\rho_{pq,p'q'}$ depends on not only the transceiver spatial separations $d_{pp'}$ and $d_{qq'}$ but also several other important factors, including 1) the angle spread $\Delta \triangleq \arcsin(R/D)$, 2) the power azimuth spread (PAS) $f(\theta)$, characterizing the scatterer density over the azimuth $\theta$ on the scatterer-ring, and 3) $\theta_T$ and $\theta_R$, determined by the azimuth positions of the transceivers. Since all these factors are environment-dependent, it is not difficult to realize that the channel correlation will change in different environments.

To apply the one-ring model for LS security analysis, one can reduce it by setting $p = p' = T_l$, $q = R_l$ and $q' = R_a$ in the case of receiver side scattering (Fig. 8.2), and then employ (8.2) to compute the corresponding correlation between the legitimate and the adversary channels;
the case of transmitter side scattering can be processed similarly by switching the roles of corresponding quantities.

With this modeling, the impacts of the angle spread $\Delta$ and adversary receiver position on channel envelope correlation $\rho_{env}$ are examined. The corresponding results are presented in Fig. 8.3.\footnote{Since most of the existing LS based applications (e.g., [45, 52]) utilize channel envelope information $|h|$, we will focus on channel envelope correlations $\rho_{env} \triangleq \frac{E[|h_{T_l, R_l}||h_{T_l, R_a}|]}{\sqrt{\text{Var}(h_{T_l, R_l})\text{Var}(h_{T_l, R_a})}}$ throughout this work, which is related to the complex channel correlation coefficient $\rho$ in (8.1) through $\rho_{env} \approx |\rho|^2$ [156].} Several important observations can be made: 1) When the scatterer ring is on the receiver side, the legitimate and the adversary channels will be quickly decorrelated by these local scatterers (Fig. 8.3a–8.3b). In such rich scattering environment, the fast spatial decorrelation assumed by existing LS techniques is valid. 2) However, when the scatterer-ring is on the transmitter side while the receivers are free from local scattering, a small angle spread $\Delta$ can induce fairly high channel correlations, as can be seen from Fig. 8.3c–8.3d. 3) In addition, by comparing Fig. 8.3c and Fig. 8.3d, it can be seen that the adversary can obtain even higher channel correlation by placing its sensor along the transmitter-to-receiver direction (corresponding to $\theta_R = 0^\circ$). For example, with a small angle spread $\Delta = 2^\circ$, the adversary can increase the channel correlation from 0.05 (Fig. 8.3c) to 0.99 (Fig. 8.3d) by changing $\theta_R$ from 90$^\circ$ to 0$^\circ$, even when the spatial separation $\delta d$ between the legitimate and the adversary receivers is 10$\lambda$, leaving security concerns.

### 8.1.2 Correlation Attack to LS

In this subsection, the correlation attack is introduced to illustrate how the attacker can exploit the high channel correlation (when exists) to impair the security of LS.

In the correlation attack, the adversary deploys $n$ ($\geq 1$) receivers, denoted by $\{R_{a_i}\}_{i=1}^n$, in the vicinity of the legitimate receiver; then based on the measured channels, denoted by $h_a = [h_{T_l, R_{a_1}}, ..., h_{T_l, R_{a_n}}]^T$ (with $T$ denoting the transpose operator) from these receivers, it constructs an estimate $\hat{h}_{T_l, R_l}$ of the legitimate channel $h_{T_l, R_l}$ through linear minimum mean square error (LMMSE) estimation.\footnote{The LMMSE estimator is optimal when the random variables involved are jointly Gaussian (often assumed in communications when the central limit theorem can be invoked), and widely adopted in practice due to its simplicity and good performance [157]. It is used here to convey the basic idea while in practice other estimators can be used as well.} Specifically, $\hat{h}_{T_l, R_l}$ is given by\footnote{In (8.5), the assumption that all the statistics are known is reasonable for certain practical situations. For example, the adversary party can deploy the transceivers in a similar environment to obtain estimates of these statistics (and build databases), or they can infer from specific physical models (e.g., the one-ring model) when these models are known to match the environment of interest well.}

$$\hat{h}_{T_l, R_l} = E[h_{T_l, R_l}] + B^T C^{-1}(h_a - E[h_a]),$$

(8.5)

where $B_{n \times 1} \triangleq \text{Cov}(h_{T_l, R_l}, h_a)$ is the correlation vector between the legitimate channel and
the adversary channels, and $C_{n \times n} \triangleq Cov(h_0, h_a)$ is the correlation matrix of the adversary channels. Several related analytical results are in order.

**Proposition 12.** The MSE of the LMMSE estimate $\hat{h}_{T_l,R_l}$ is given by $\frac{\det(\Gamma)}{\det(C)}$, where $\Gamma = \begin{bmatrix} A & B^T \\ B & C \end{bmatrix}$ and $A_{1 \times 1} \triangleq Var(h_{T_l,R_l})$ is the variance of the legitimate receiver channel.

*Proof.* Please see Appendix F.1.

*Remark:* In the special case of $n = 1$, it can be verified that the normalized (with respect to the variance $A$ of $h_{T_l,R_l}$) MSE of $\hat{h}_{T_l,R_l}$ is $1 - \rho^2$ (with $\rho$ denoting the correlation coefficient between the legitimate and the adversary channels), which indicates that higher channel correlation allows the attacker to obtain finer estimate and thereby causes more severe threats to the legitimate LS.

**Proposition 13.** The estimator $\hat{h}_{T_l,R_l}$ is always no worse than that based on any subset of $\{h_{T_i,R_i}\}_{i=1}^n$ with $k(< n)$ adversary sensors.

*Proof.* Please see Appendix F.2.

Actually, not only that $\hat{h}_{T_l,R_l}$ becomes more accurate with more adversary receivers deployed, it can be also shown that in some circumstances with the presence of sufficient high
correlation between the legitimate and the adversary channels, the adversary is even capable of perfectly reconstructing the legitimate channel by increasing the number of adversary receivers, as given in the following corollary.

**Corollary 2.** Assume that the correlation between any adversary and the legitimate channels is $\rho$, and the correlation between any two adversary channels is $\rho'$. Then, if $\rho^2 > \rho'$, there exists an $n = \lceil \frac{1-\rho'}{\rho^2-\rho'} \rceil$, such that the MSE of the attacker’s estimate can be driven down to zero when employing $n$ adversary receivers.

**Proof.** Please see Appendix F.3. \qed

To further illustrate Corollary 2, numerical result for $\rho = 0.9$ and $\rho' = 0.8$ is presented in Fig. 8.4. From Fig. 8.4, it can be seen that, 8 to 10 adversary receivers will result in satisfactory estimation quality to the attacker, and by further increasing to 20 adversary receivers, the adversary can even achieve perfect estimation.

Then we move one step further to consider a more practical example, where the channel correlations among all channels are assumed to be determined by the one-ring model with transmitter side scattering and the adversary receivers are deployed along the transmitter-to-receiver direction (i.e., $\theta_R = 0^\circ$ in (8.2)). The corresponding normalized MSE’s of the attacker’s estimate for different numbers of adversary sensors are given by Proposition 12 and presented in Fig. 8.5. As shown in Fig. 8.5, when $\Delta$ is small ($\Delta = 6^\circ$), a single adversary receiver placed around 5 wavelengths away from the legitimate receiver is able to achieve a target normalized MSE 0.05. If the adversary has two collaborative receivers, both of them may be put at least 10 wavelengths away, and for eight adversary receivers the target is still achieved even if the spatial separation is 20 wavelengths.

These results clearly indicate that the commonly believed half-wavelength separation may not be sufficient to protect the LS itself in certain environments. For a more clear understanding on the suitable guard zones for LS in different environments, a more comprehensive studies on

---

Figure 8.4: A numerical example (averaged over 10000 Monte Carlo runs).
channel correlation will be conducted in the next section.

8.2 Key Channel Factors/Models for LS Security

In this section, various wireless channel correlation models are investigated and several key factors that have substantial impacts on channel correlation and LS security are identified.

8.2.1 Power Azimuth Spectrum and the Azimuth Spread

Besides the angle spread $\Delta$ discussed in Section 8.1.1, the PAS $f(\theta)$ (c.f. (8.2)), which describes the scatterer density over the azimuth $\theta$, is another important factor for channel correlation and LS security. In the previous discussion, uniform PAS ($f(\theta) = 1/2\pi$) is assumed; various other PAS’s are also proposed in literature such as the cosine function PAS [153], the truncated uniform PAS [158], the truncated Gaussian PAS [159, 160], the von-Mises PAS [155] and the truncated Laplacian PAS [161]. These single-mode scatterer distributions (i.e., scatterers mostly concentrate around a mean azimuth $\bar{\theta}$) are compared in Fig. 8.6, and can be easily extended to multi-mode ones when multiple clusters of scatterers exist [161].

Figure 8.6: Comparison of different PAS’s with $AS = 0.25$ and $\bar{\theta} = 0^\circ$. 

Figure 8.5: Achievable normalized MSE of $n$ adversary receivers aligned in a line ($\Delta = 6^\circ$).
The azimuth spread (AS)\textsuperscript{5} is a generic metric to measure the concentrations of scatterers for different PAS's, which is defined as \[ AS \triangleq \sqrt{1 - \left| F_1 \right|^2/\left| F_0 \right|^2}, \] (8.6)

where \( F_n = \int_0^{2\pi} f(\theta) \exp(jn\theta) d\theta \) is the \( n \)th complex Fourier coefficient of \( f(\theta) \). The AS ranges from 0 to 1 where \( AS = 0 \) (e.g., when \( \kappa \to \infty \) in von-Mises PAS \( f(\theta) = \exp(\kappa \cos(\theta - \bar{\theta})) / 2\pi I_0(\kappa) \), with \( \theta \in (-\pi, \pi) \) and \( I_0(\cdot) \) denoting the modified Bessel function of the first kind) corresponds to signal incidence from a single direction and \( AS = 1 \) (e.g., when \( \kappa = 0 \) in von-Mises PAS) corresponds to all-around arrivals.

Fig. 8.7 presents the spatial channel correlation functions for different PAS's of the same AS, based on the one-ring model (8.2). As it can be seen that 1) channel correlation is not very sensitive to the particular forms of PAS but is mainly determined by the corresponding AS, and that 2) a smaller AS always leads to higher channel correlation. This implies that much larger guard zones are needed for LS in the environments with highly concentrated scatterers.

### 8.2.2 Rician Factor

In addition to the (random) diffusion component caused by the scattering effect (as considered in (8.2)), a wireless channel may also contain a (deterministic) line-of-sight (LOS) component. For such cases, the so-called Rician factor, denoted by \( K \), is defined as the power ratio between these two components, and correspondingly a space-time correlation \( \tilde{\rho} \) is defined in [155], given by

\[ \tilde{\rho} = \frac{\left| F_1 \right|^2}{\left| F_0 \right|^2}, \]

\[ \frac{\left| F_1 \right|^2}{\left| F_0 \right|^2} \]

---

\textsuperscript{5}The azimuth spread is also called angular spread in [162], which is determined by the angular domain scatterer distribution, and should not be confused with the angle spread \( \Delta \) defined earlier, which is determined by the scatterer-ring size and transmitter-to-receiver distance.
\[ \tilde{\rho}_{pq,p'q'} \triangleq \frac{\mathbb{E}[h_{pq}h_{p'q'}^*]}{\sqrt{\mathbb{E}[|h_{pq}|^2]\mathbb{E}[|h_{p'q'}|^2]}} = \rho_{pq,p'q'}^{\text{DIF}} + \rho_{pq,p'q'}^{\text{LOS}}, \quad (8.7) \]

where the space-time correlation for the diffusion component \( \rho_{pq,p'q'}^{\text{DIF}} \) can be computed by (8.2) with a scaling factor \( 1/(1 + K) \), and that for the LOS component is given by

\[ \rho_{pq,p'q'}^{\text{LOS}} = \frac{K}{1 + K} \exp \left\{ \frac{j2\pi}{\lambda} [d_{pp'} \cos(\theta_T) - d_{qq'} \cos(\theta_R)] \right\}. \quad (8.8) \]

Based on (8.7), it can be verified that large Rician factor induces high space-time correlation, which seemingly implies a severe vulnerability of LS when a strong LOS component exists. Considering this, existing LS based security applications are investigated as to how the LOS component is handled. It is found that, in LS based secret key generation algorithms, the LOS component is removed from the channel measurement before the key generation process \([45, 47, 48, 163, 164]\), and that, in the location distinction algorithms, the LOS effect is also removed implicitly by comparing the difference between two channel measurements to the standard deviation (instead of the channel magnitude) \([52–54, 56, 165]\). Therefore, the existence of the LOS component will not have a significant impact on the security of these LS based applications. In the following, only the diffusion part will be considered.

### 8.2.3 Directive Antenna

In practice, directive antenna is often used to enhance communication performance by suppressing signals from unwanted directions. The gain of a directive antenna in azimuth \( \theta \) is
characterized by its radiation pattern $G(\theta)$, which is parameterized by the main lobe direction $\theta_G$ and the 3-dB antenna beamwidth $\theta_{3dB}$.

When the adversary employs the same directive antenna as the legitimate receiver, the corresponding channel correlation is given by [161]

$$
\rho = \int_{-\pi}^{\pi} \exp \left\{ \frac{2\pi j}{\lambda} \delta d \cos(\theta_R - \theta) \right\} f(\theta) G(\theta) d\theta, \tag{8.9}
$$

where receiver side scattering is assumed. It is worth noting from (8.9) that, mathematically, the PAS $f(\cdot)$ and the antenna radiation pattern $G(\cdot)$ have equivalent impacts on channel correlation.

Based on (8.9), the spatial correlation functions with different radiation patterns are compared in Fig. 8.8, where a truncated uniform PAS with $\theta_{\text{max}} = 10^\circ$ and $\theta_{\text{min}} = -10^\circ$ is assumed. It can be seen that highly directional antennas (with small $\theta_{3dB}$) can induce large channel correlations. For example, when the adversary-to-legitimate receiver separation $\delta d = 4\lambda$, using both bell-shape [161] and flat-top [166] directive antennas with $\theta_{3dB} = 5^\circ$ will boost the channel correlation from near zero to about 0.5. (Similar trends are also observed for other PAS’s as well.)

An intuitive explanation for this correlation boosting phenomenon of directive antenna is as follows: First notice that channel decorrelation is essentially caused by that the signal phase shifts due to different scatterers are independent; the directive antenna will suppress the signals reflected by those scatterers in unwanted direction (which equivalently leads to a more concentrated PAS) and hence reduces the randomness in scattering, inducing high channel correlation. In general, channel correlation boosting effect appears only when directive antenna reduces the angular domain spread of effective scatterers, i.e., the scatterers illuminated by the directive antenna, as shown in Fig. 8.9, which explains that in Fig. 8.8, channel correlation is significantly enhanced only when $\theta_{3dB} < (\theta_{\text{max}} - \theta_{\text{min}})$.

### 8.2.4 Other Models for Different Scattering Environments

Different scattering environments other than that assumed by the one-ring model exist in practice, and several other channel models will be studied in this subsection to account for these cases. For these models, previous conclusions as to angle spread, PAS/AS, Rician factor and directive antenna in one-ring model carry over when applicable.

#### Two-Ring Models

In both indoor and outdoor environments, both communication ends may be enclosed by local scatterers. In these cases, two-ring models [60, 61, 167–169] can be employed to characterize the corresponding channel correlation. With different assumptions on signal propagation, both the
single-bounce and the double-bounce two-ring models are proposed in literature.

In the single-bounce two-ring model, it is assumed that the received signals are reflected by either the transmitter side or the receiver side scatterers, as depicted in Fig. 8.10. With this assumption, the single-bounce model is in fact a weighted superposition of two one-ring models with corresponding scatterer-rings on the transmitter and the receiver sides, respectively [61]. The correlation due to transmitter side scatterers is given by

$$\rho^{(SBT)}(\theta', \varphi) = \int_{-\pi}^{\pi} \exp \left\{ \frac{2\pi j}{\lambda} [\delta d \cos(\theta_R - \varphi)] \right\} f_T(\theta') G_T(\theta') d\theta', \quad (8.10)$$

where $\theta'$ and $\varphi$ admit similar relations in (8.3) and (8.4). The correlation due to receiver side scatterers is given by

$$\rho^{(SBR)}(\theta') = \int_{-\pi}^{\pi} \exp \left\{ \frac{2\pi j}{\lambda} [\delta d \cos(\varphi - \theta)] \right\} f_R(\theta) G_R(\theta) d\theta. \quad (8.11)$$

The overall correlation is given by

$$\rho^{(SB)} = \left( \eta_{SBT} \cdot \rho^{(SBT)} + \eta_{SBR} \cdot \rho^{(SBR)} \right), \quad (8.12)$$

where $\eta_{SBT}$ and $\eta_{SBR}$ represent the strengths of the reflected signals from the two scatterer-rings, respectively, and admit $\eta_{SBT} + \eta_{SBR} = 1$.

In the double-bounce two-ring model [60], wireless signals get reflected and scattered at both the transmitter side and receiver side scatterers; nevertheless, it can be verified that, for the purpose of LS security assessment where only one transmitter is considered, the double-bounce two-ring model reduces to the one-ring model.
Elliptical Ring Model

In the elliptical scatterer-ring model, an elliptical scatterer-ring encloses both the transmitter and the receivers, as depicted in Fig. 8.11. This model may be applied to office environment where the two communication ends are not far from each other and surrounded by common scatterers nearby. Denoting the major and minor radii by $a$ and $b$, respectively, the corresponding channel correlation is given by [62] (and the discussions in Section 8.2.3)

$$\rho^{(E)} = \int_{-\pi}^{\pi} \exp\left\{ \frac{2\pi f}{\lambda} \left[ \delta d \cos(\theta_R - \alpha_R) \right] \right\} f_E(\theta)G_T(\alpha_T)G_R(\alpha_R) d\theta,$$

(8.13)

where the expression of $\alpha_R$ as a function of $\theta$ is determined by the geometry shown in Fig. 8.11.

The channel correlation under elliptical scatterer-ring modeling is shown in Fig. 8.12. It can be seen that a narrower elliptical scatterer-ring (i.e., smaller $a$ with fixed center-to-focus distance $\sqrt{a^2 - b^2} = 1$) will induce higher channel correlation.

\footnote{The subscripts $T$ and $R$ of $f (G)$ denote the transmitter and receiver side PAS (antenna pattern), respectively.}

Figure 8.11: Elliptical scatterer-ring model.

Figure 8.12: Correlation comparisons in elliptical scatterer-ring model.

Figure 8.13: Far scatterer-ring model.
Table 8.1: Important factors for LS security. (O: one-ring, T: two-ring, E: elliptical scatterer-ring, F: far scatterer-ring)

<table>
<thead>
<tr>
<th>Applicable models</th>
<th>Favorable value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angle spread $\Delta$ ($\Delta_F$)</td>
<td>O, T (F)</td>
</tr>
<tr>
<td>Scatterer distribution $AS$</td>
<td>O, T, E, F</td>
</tr>
<tr>
<td>Rician factor $K$</td>
<td>O, T, E, F</td>
</tr>
<tr>
<td>Directive antenna $\theta_{3dB}$</td>
<td>O, T, E, F</td>
</tr>
<tr>
<td>Eccentricity $\sqrt{a^2 - b^2}/a$</td>
<td>E</td>
</tr>
<tr>
<td>Adversary’s angular position $\theta_R$</td>
<td>O, T, E, F</td>
</tr>
</tbody>
</table>

**Far Scatterer-Ring Model**

Far scatterer-ring model in which the scatterers that are distant from both communication ends, as depicted in Fig. 8.13, has been considered in [170], and it is mentioned that the correlation function due to far scatterers is mathematically the same as that due to local scatterers and only the scatterers are centered around a different position. Based on this principle, the correlation function due to far scatterers can be derived as

$$\rho^F(\theta) = \int_{-\pi}^{\pi} \exp\left\{\frac{2\pi j}{\lambda} [\delta d \cos(\theta_R - \alpha_R)]\right\} \cdot f_F(\theta) G_T(\alpha_T) G_R(\alpha_R) d\theta,$$

(8.14)

where the angle spread for the far scatterer-ring is defined as $\Delta_F \triangleq \arcsin(R_F/D)$, and relevant parameters can be found in Fig. 8.13.

Based on (8.14), channel correlations due to far scatterers are examined in Fig. 8.14. It can be seen from Fig. 8.14a that for the same x-position of the scatterer-ring center, higher correlations are observed for larger $H_F$ (determined by $\gamma_T$ and $\gamma_R$), and from Fig. 8.14b–8.14d that small angle spread, highly concentrated PAS and directive antenna pattern will induce high channel correlation, as in the near-scatterer case.

### 8.2.5 A Generic Channel Correlation Model for LS

Based on the previous discussions, the key channel factors and models for LS security assessment are summarized in Table 8.1, together with the corresponding most favorable values in terms of LS security. Based on the obtained understanding and insights, a generic channel correlation model that includes the security implications of all these factors and models is given as follows:
Figure 8.14: Spatial correlations of two receivers due to far scatterers ($\theta_T = 0$, $\theta_R = 0$).
\[
\rho = \eta_{SBT} \cdot \rho^{(SBT)} + \eta_{SBR} \cdot \rho^{(SBR)} + \eta_{E} \cdot \rho^{(E)} + \eta_{F} \cdot \rho^{(F)},
\]
where the sub-model coefficients admit \( \eta_{SBT} + \eta_{SBR} + \eta_{E} + \eta_{F} = 1 \) and \( \rho^{(SBT)}, \rho^{(SBR)}, \rho^{(E)}, \rho^{(F)} \) are given by (8.10), (8.11), (8.13) and (8.14), respectively. Some explanations are in order. First, the LOS component is omitted, since it will not change the adversary’s attacking performance, as discussed in Section 8.2.2. The one-ring model is a special case of the single-bounce two-ring model captured here, and so is the double-bounce two-ring model as far as LS security assessment is concerned. As will be seen in the next section, this weighted sum form provides flexibility in modeling channel correlations in various environments of interest with properly chosen weighting coefficients, either by selecting the most suitable model (as in Scenario I), or by an appropriate combination of roughly independent sub-models (as in Scenario II and III).

### 8.3 Simulation and Numerical Results

As can be noted from the above discussions, the correlation of wireless channels varies substantially depending on the scattering environment and hence, the commonly believed half-wavelength cannot ensure LS security universally. As will be shown in this section, in many scenarios, much larger guard zone may need to be deployed around legitimate devices to create sufficient channel decorrelation for LS security assurance.

Specifically, the guard zone sizes for three different typical wireless communication scenarios are numerically explored based on the discussed channel models.\(^7\) The performances of two LS based security mechanisms, secret key generation \cite{keygen} and location distinction \cite{location}, are investigated under the correlation attack when guard zones of different sizes are deployed. To account for various physical environments, multiple combinations of parameters are chosen for each of the three scenarios. In all simulations, the legitimate and the adversary channels assume Rayleigh fading with correlation given by (8.15) based on the corresponding environment parameters. We focus on presenting the results for one adversary receiver case to convey the basic idea; when multiple collaborative adversary receivers are deployed, even larger guard zones are needed, in view of Proposition 13.

Considering that in practice different cryptographic algorithms and detection thresholds may be used for LS based secret key extraction and location distinction, two security levels, Lv1 and Lv2, are considered in this work. For Lv1 (Lv2) security, it is assumed that the promised

\(^7\)In this work, we focus on narrowband and single antenna cases and the extensions to wideband and multiple antenna remains future work.
Table 8.2: Scenario I

<table>
<thead>
<tr>
<th>Model</th>
<th>Elliptical-ring model ($\eta_E = 1$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a : b$</td>
<td>4 : 1, 4 : 2, 4 : 3</td>
</tr>
<tr>
<td>PAS</td>
<td>uniform, von-Mises ($\kappa = 10, 50; \bar{\theta} = 0^\circ, 90^\circ, 180^\circ$)</td>
</tr>
<tr>
<td>$G_R$</td>
<td>omni, $\theta^{(R)}_{3dB} = 40^\circ$;</td>
</tr>
</tbody>
</table>

security by the LS is thwarted if the normalized MSE$^8$ of the adversary’s estimated channel in the location distinction application is below 0.1 (0.5) or more than 90% (50%) secret key bits are inferred by the adversary. For secret key extraction, the algorithm in [45] is implemented where $1 \times 10^4$ samples are generated for each channel and an excursion of length 4 is used. Finally, the (empirical) outage probability $P_{\text{out}}(\delta d)$, defined as

$$P_{\text{out}}(\delta d) \triangleq \frac{\text{number of insecure environments}}{\text{total number of considered environments}}, \quad (8.16)$$

is employed as the metric for LS security assessment, and a non-zero $P_{\text{out}}(\delta d)$ implies the existence of environment(s) where the LS application is insecure when the guard zone size is $\delta d$.

A typical indoor scenario (Scenario I) is considered first, where both the legitimate transceivers and the adversary receiver are in the same office. In such a scenario, both communication ends are surrounded by common scatterers, and hence the elliptical scatterer-ring model can be used

$^8$Note that in the location distinction application [52], the channel difference $||\hat{h} - \bar{h}||$ (with $\bar{h}$ the empirical average channel) is compared with channel standard deviation $\sigma_h$ for location change detection. When the adversary can obtain an estimate of the legitimate channel with small MSE, it can launch the mimicry attack [56] so as to spoof the detector that it is located at the same position as the legitimate transceiver.
Table 8.3: Scenario II

<table>
<thead>
<tr>
<th>Model</th>
<th>Two-ring model ($\eta_{SBT} = 0.9, \eta_{SBR} = 0.1$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta$</td>
<td>$2^\circ, 5^\circ, 10^\circ, 20^\circ$</td>
</tr>
<tr>
<td>$PAS_T$</td>
<td>uniform, von-Mises ($\kappa = 10, 50; \bar{\theta} = 0^\circ, 90^\circ, 180^\circ$)</td>
</tr>
<tr>
<td>$PAS_R$</td>
<td>uniform, von-Mises ($\kappa = 10, 50; \bar{\theta} = 0^\circ$)</td>
</tr>
<tr>
<td>$G_T$</td>
<td>omni, $\theta_{3dB}^{(T)} = 40^\circ$;</td>
</tr>
<tr>
<td>$G_R$</td>
<td>omni, $\theta_{3dB}^{(R)} = 40^\circ$;</td>
</tr>
</tbody>
</table>

to characterize the corresponding channel correlations. To account for various physical environments in this scenario, 42 different parameter combinations are considered, as summarized in Table 8.2. For example, 3 different $a : b$ and $(1 + 2 \times 3)$ different PAS’s are used to account for different office sizes and scatterer distributions, respectively.

Fig. 8.15 and Fig. 8.16 show the outage probabilities for LS based location distinction and secret key extraction, respectively. It can be seen that the commonly believed safe-distance $\lambda/2$ is not sufficient to secure the LS applications for all the cases. For example, as shown in Fig. 8.15, when $\delta d = \lambda/2$, in more than 50% of the 21 considered cases, the adversary can obtain an estimate of the legitimate LS with normalized MSE less than 0.1 and thus defeats the Lv1 security requirement of location distinction; it becomes even worse (i.e., larger $P_{out}$) when both the legitimate and adversary receivers adopt directive antennas (in the rest 21 cases). Similar observations can be made in Fig. 8.16 for LS based secret key extraction as well. In fact, the results in Fig. 8.15 and Fig. 8.16 suggest that a guard zone of size about $\delta d = 19\lambda$ is needed to achieve Lv1 security with zero outage probability for LS. For the more demanding Lv2 security, even larger guard zones are required.

In the second scenario (Scenario II), the transmitter is inside the office (with rich scattering) while both the legitimate and the adversary receivers are in the hallway (with much less scattering). A single-bounce two ring model with weighting coefficients $\eta_{SBT} = 0.9$ and $\eta_{SBR} = 0.1$ is employed to characterize the channel correlations, and the corresponding parameter settings are given in Table 8.3 with a total of 336 combinations. Again, a guard zone of size $\delta d = \lambda/2$ cannot ensure LS security, as shown in Fig. 8.17 and Fig. 8.18. Instead, a guard zone of size $\delta d = 12\lambda$ has to be deployed to achieve Lv1 security. Although a smaller guard zone is required in this scenario for Lv1 security, it can be seen by comparing Figs. 8.15–8.16 and Figs. 8.17–8.18 that Scenario II requires a larger guard zone for Lv2 security as compared to Scenario I.

The last scenario (Scenario III) assumes a base station-to-mobile user communication where
Table 8.4: Scenario III

<table>
<thead>
<tr>
<th>Model</th>
<th>One-ring &amp; far scatterer-ring models ($\eta_{SBR} = 0.8$, $\eta_F = 0.2$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta$ and $\Delta_F$</td>
<td>$2^\circ$, $5^\circ$, $10^\circ$, $20^\circ$</td>
</tr>
<tr>
<td>$PAS_R$</td>
<td>uniform, von-Mises ($\kappa = 10$, $\bar{\theta} = 0^\circ$, $90^\circ$, $180^\circ$)</td>
</tr>
<tr>
<td>$PAS_F$</td>
<td>uniform, von-Mises ($\kappa = 10$, $\bar{\theta} = 0^\circ$, $90^\circ$, $180^\circ$)</td>
</tr>
<tr>
<td>Far scatterers position</td>
<td>$\gamma_T = 45^\circ$, $\gamma_R = 90^\circ$, $135^\circ$</td>
</tr>
<tr>
<td>$G_R$</td>
<td>omni, $\theta_{3dB}^{(R)} = 40^\circ$;</td>
</tr>
</tbody>
</table>

the transmitter is assumed high raised (with less scattering) and the legitimate and adversary receivers are surrounded by scatterers, and far scatterers exist as well. A weighted combination of one-ring and far scatterer-ring models is employed to characterize the channel correlation in such scenario, and the corresponding parameter settings are given in Table 8.4 with a total of 3136 combinations. As it can be seen from Fig. 8.19 and Fig. 8.20 that $\delta d = 5\lambda$ is required for Lv1 security and Lv2 again requires a larger guard zone. In this outdoor scenario, the dense local scatterers ($\eta_{SBR} = 0.8$) around the receivers decorrelate the legitimate and the adversary channels fairly quickly and thus provide better security protection for LS (i.e., smaller guard zones are needed), as compared to the previous two indoor scenarios.

8.4 Experimental Verification

In this section, experiment results obtained from a Universal Software Radio Peripheral (USRP) platforms and GNURadio prototype implementation are presented. It is worth pointing out
Table 8.5: Indoor Experiment

<table>
<thead>
<tr>
<th>δd</th>
<th>$\hat{\rho}$ (σ$\hat{\rho}$)</th>
<th>NMSE (σNMSE)</th>
<th>$\bar{\xi}$ (σ$\bar{\xi}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5λ</td>
<td>0.86 (0.06)</td>
<td>0.24 (0.1)</td>
<td>77% (9%)</td>
</tr>
<tr>
<td>3.3λ</td>
<td>0.77 (0.06)</td>
<td>0.39 (0.1)</td>
<td>74% (10%)</td>
</tr>
</tbody>
</table>

that high channel correlation has already been observed, even when the spatial separation is more than $\lambda/2$, in the context of MIMO systems [58]. The experiment results and discussions presented here focus on the LS context and aim at providing real-world justifications to the previous analysis.

In our experiments, the carrier frequency is 2.4 GHz with the corresponding wavelength 12.5 cm and the channel sampling rate is 100 samples/sec. Both indoor and outdoor experiments are conducted. In each experiment, 40 pairs of legitimate and adversary channels are recorded for two different spatial separations $\delta d = 1.5\lambda$ and $\delta d = 3.3\lambda$, respectively. Based on these channel measurements, the normalized MSE between the legitimate and the adversary channels are computed to assess the LS based location distinction, and the match rates $\xi$ between the secret keys generated from the legitimate channel and the adversary channel (by using the level-crossing algorithm [45]) are computed as well to assess the LS based secret key extraction.

The setting of the indoor experiment is depicted in Fig. 8.21a. In this experiment, the transmitter is placed in an office room (Fig. 8.21b) with ample scatterers around, while the two receivers (with one of them serving the role of the adversary sensor) are located at the end of the hallway with a large glass window behind, receiving less scattering. Fig. 8.22 presents one (out of the 40) measured channel samples for $\delta d = 3.3\lambda$, and it can be seen that, even when the corresponding spatial separation is substantially larger than half-wavelength, the legitimate and the adversary channels can still exhibit high correlation. When the adversary further employ

![Figure 8.19: Location distinction in Scenario III. (“O-”: Omni-directional; “D-”: Directive)](image1)

![Figure 8.20: Secret key generation in Scenario III. (“O-”: Omni-directional; “D-”: Directive)](image2)
Figure 8.21: Indoor experiment.

Figure 8.22: Channel measurements in the indoor experiment when $\delta d = 3.3\lambda$. 
LMMSE estimation in (8.5), the corresponding average values of channel correlation ($\bar{\rho}$), normalized MSE ($\bar{NMSE}$) and key match rate ($\bar{\xi}$) over the collection of all the measurements are summarized in Table 8.5, where the numbers in the parentheses are the corresponding sample standard deviations. It can be seen that even when the adversary receiver is separated from the legitimate one by more than half-wavelength, fairly high correlations around 0.8 are observed; in these cases, the attacker can construct an estimate of the legitimate LS with normalized MSE around 0.3 and can successfully infer around 75% legitimate secrecy bits.

The setting of the outdoor experiment is depicted in Fig. 8.23, where the transmitter is placed behind a building pillar while the two receivers are placed in a large open lawn (without much scatterers nearby). The corresponding experiment results are summarized in Table 8.6. Again, it can be observed that the adversary can recover the legitimate LS and the corresponding generated secret bits with substantial fidelity.

It can be seen that in the environments of these two experiments, if guard zones with size of only half-wavelength are deployed, a large portion of the secrecy of the legitimate LS will be inferred by the attacker and hence significantly weaken the promised security protection to the legitimate devices. This observation is consistent with our previous analysis, and in fact, the one-ring model may be employed to provide an intuitive explanation: In both experiments, the transmitter is surrounded by relatively rich scatterers while the receivers experience much less scatterering; in such cases, two wireless channels will be highly correlated given a small angle spread. In such environments, much larger guard zones are needed to protect the legitimate LS.

### 8.5 Conclusions

After illustrating potential vulnerabilities of LS through correlation attack when high channel correlation exists, several key factors that have important influence of LS security are identified

---

9The exact amount of performance degradation to a LS based security scheme depends on the specific implementation and is beyond the scope of this work.
through a comprehensive investigation of well-established channel models. With the obtained understanding and insights, a generic model characterizing the spatial correlation between the legitimate and the adversary channels is developed to explore proper guard zone sizes for LS. Both our numerical and experimental results indicate that the spatial channel correlation varies for different wireless environments. In particular, the commonly believed half-wavelength decorrelation assumption is valid mainly in environments with rich scattering; while in poor scattering environments, the legitimate and the adversary channels will decorrelate much slower over space. These findings suggest that in practice, more careful investigations on channel correlation for the specific environment of interest must be conducted before a confident deployment of LS based security mechanisms.

<table>
<thead>
<tr>
<th>$\delta d$</th>
<th>$\bar{\rho}$ ($\sigma_{\rho}$)</th>
<th>$\text{NMSE}$ ($\sigma_{\text{NMSE}}$)</th>
<th>$\bar{\xi}$ ($\sigma_{\xi}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1.5\lambda$</td>
<td>0.91 (0.04)</td>
<td>0.18 (0.06)</td>
<td>86% (7%)</td>
</tr>
<tr>
<td>$3.3\lambda$</td>
<td>0.83 (0.06)</td>
<td>0.31 (0.09)</td>
<td>74% (9%)</td>
</tr>
</tbody>
</table>
Chapter 9

Summary and Future Work

9.1 Conclusions

While bringing unprecedented convenience and productivity to our lives, wireless systems have also exhibited vast vulnerabilities for the adversaries to explore, leaving its security an ever present concern. In this dissertation, we investigated some important and timely problems in wireless security.

In Chapter 2 and 3, two effective defense schemes are developed to detect Byzantine attackers in CR networks. In the first scheme, hidden Markov model (HMM) is employed to characterize users’ sensing behaviors, and a new HMM estimation algorithm that can jointly estimate multiple correlated HMMs using only unlabeled data is developed. Subsequently, this algorithm is applied to separate malicious users from honest ones, providing an effective defense against the Byzantine attack. In Chapter 3, another countermeasure to the Byzantine attack, termed Conditional Frequency Check (CFC), is proposed, in which two natural but effective CFC statistics that explore the second-order properties of the underlying Markov spectrum model are constructed. Both proposed methods can defend against malicious users that flip local sensing results in both directions and do not require any prior knowledge of the true spectrum status.

Our following efforts were put on the study of dynamic security competitions where the attacker can adaptively change its strategy. In Chapter 4, a dynamic mobile jamming/anti-jamming competition is considered. For this problem, new spectral quantities, single- and multi-weighted Cheeger constants and corresponding eigenvalue variants, are proposed based on spectral graph theory. As compared to existing literature, our approach offers not only more effective mobility control strategy but also certain level of performance assurance. Moving further, we studied dynamic security games with information asymmetry in Chapters 5–7, where the defender and the attacker can not only adapt to the opponent’s current strategy but also
think ahead and conduct foresighted planning. Particularly, in Chapter 5, two new MARL algorithms, termed minimax-PDS and WoLF-PDS, are proposed to expedite the learning and adaptation of the defender when information advantage exists at the defender side. Through analysis, we show that these two algorithms are provably convergent and rational, respectively. The studies in Chapter 6 and Chapter 7 are devoted to the cases where the defender has uncertainties about the adversary. Particularly, a joint RL and type identification algorithm and a Bayesian Nash-Q learning algorithm are proposed for the cases of zero-sum and non-zero-sum security games, respectively. By using the proposed algorithms, the defender can gradually infer the unknown information based on the observed actions from the opponent and adjust its defense strategy accordingly.

Besides developing new algorithms, the recently advocated link-signature based security mechanisms are reexamined in Chapter 8. In particular, existing wireless channel models are revisited, with a focus on their security implications to link-signature. Several key factors that have significant influence on link-signature security have been identified, and based on the obtained understanding, a generic channel correlation model is developed to facilitate the study on the guard zone deployment in typical indoor and outdoor communication scenarios. Our results suggest that the commonly believed half-wavelength decorrelation assumption does not always hold, and thus in practice, more careful investigation on channel correlation for the specific environment of interest must be conducted before a confident deployment of link-signature based security mechanisms.

9.2 Achievements

We have tackled some open and interesting problems in wireless security, and our contributions include: 1) enhancing the security and robustness of collaborative spectrum sensing in CR networks, 2) developing new metrics and algorithms for dynamic security games, and 3) evaluation of the dependability of link-signature for security provisioning. Part of this PhD work has been published or submitted for publication [171–184], which includes three published journal papers, seven conference papers, one poster, two journal papers under review and one submitted conference paper.

9.3 Future Works

Finally we list below some of the interesting topics that deserve further study.

- Defending against more advanced Byzantine attackers (c.f. Chapter 2 and 3) with adaptive time-varying flipping probabilities, or that can intelligently construct falsified reported sequences based on the defense mechanisms.
• Extension of the algorithms proposed in Chapter 5 beyond the zero-sum setting, incorporating the PDS-learning principle into other relevant MARL algorithms (such as Nash-Q and FoF-Q [33]), and considering further applications in similarly structured problems.

• Exploring further applications of the algorithms proposed in Chapter 6 and Chapter 7.

• Further examination on the performance of the proposed approach in Chapter 7 in real IDSs and consideration of more intelligent attackers who may explicitly attempt to misguide the IDS.
REFERENCES


163


APPENDICES
Appendix A

A.1

In this appendix, it will be shown that \( \log P(O|\bar{\lambda}) \geq \Phi(\bar{\lambda}|\lambda_n) \) \( \forall \bar{\lambda} \) and \( \log P(O|\lambda_n) = \Phi(\lambda_n|\lambda_n) \) for convergence analysis. To see these, notice that for any \( \bar{\lambda} \)

\[
\log P(O|\bar{\lambda}) - \log P(O|\lambda_n) = \sum_{q \in Q} \sum_{m \in M} P(O|q, m, \bar{\lambda})P(q, m|\bar{\lambda}) \frac{P(q, m|O, \lambda_n)}{P(q, m|O, \lambda_n)} - \log P(O|\lambda_n)
\]

\[
\geq \sum_{q \in Q} \sum_{m \in M} P(q, m|O, \lambda_n) \log \frac{P(O|q, m, \bar{\lambda})P(q, m|\bar{\lambda})}{P(q, m|O, \lambda_n)} - \log P(O|\lambda_n)
\]

\[
= \sum_{q \in Q} \sum_{m \in M} P(q, m|O, \lambda_n) \log \frac{P(O|q, m, \bar{\lambda})P(q, m|\bar{\lambda})}{P(q, m|O, \lambda_n)P(O|\lambda_n)}
\]

\[
= \Delta(\bar{\lambda}|\lambda_n), \quad (A.1)
\]

where the Jensen’s inequality is applied. Combining with (2.15), \( \log P(O|\bar{\lambda}) \geq \Phi(\bar{\lambda}|\lambda_n) \) follows. In addition, it can be readily verified that \( \Delta(\lambda_n|\lambda_n) = 0 \) from (2.14). Substituting this in (2.15) leads to \( \log P(O|\lambda_n) = \Phi(\lambda_n|\lambda_n) \).

A.2

Derivations for equations (2.16)–(2.19) are presented here. To see (2.16), we can start from (2.8):
\[
c_m = \frac{1}{L} \sum_{j=1}^{L} \frac{P(O, m^j = \bar{m}|\lambda')}{P(O|\lambda')} \propto \frac{1}{L} \sum_{j=1}^{L} P(O, m^j = \bar{m}|\lambda') = \frac{1}{L} \sum_{l=1}^{L} \left( c'_m \sum_{i=0}^{S-1} \alpha_{i, l}^{l, \bar{m}}(T) \right), \quad (A.2)
\]

where the definitions of \(\alpha_{i, l}^{l, \bar{m}}(t)\) with \(t = T\) and \(c'_m\) are applied in the last step. To see (2.17), we can start from (2.9) and apply definition of \(\beta_{j, \bar{m}}(1)\):

\[
\pi_i = P(O, q_i = i|\lambda')/P(O|\lambda') \propto P(O, q_0 = i|\lambda')
= \sum_{\bar{m}=1}^{N} P(O, m^l = \bar{m}, q_0 = i|\lambda')
= \sum_{\bar{m}=1}^{N} \left( P(m^l = \bar{m}|\lambda') P(q_0 = i|m^l = \bar{m}, \lambda') \cdot P(O|m^l = \bar{m}, q_0 = i, \lambda') \right)
= \sum_{\bar{m}=1}^{N} \left( c'_m \alpha_{i}^{l, \bar{m}}(0) \sum_{j=1}^{S} P(O, q_1 = j|m^l = \bar{m}, q_0 = i, \lambda') \right)
= \sum_{\bar{m}=1}^{N} \left( c'_m \alpha_{i}^{l, \bar{m}}(0) \sum_{j=1}^{S} a_{i,j} \beta_{j, \bar{m}}(1) \right). \quad (A.3)
\]

Similarly, it follows from (2.11) that

\[
a_{i,j} \propto \sum_{t=1}^{T} P(O, q_{t-1} = i, q_t = j|\lambda')
= \sum_{t=1}^{T} \sum_{\bar{m}=1}^{N} P(m^l = \bar{m}|\lambda') P(O, q_{t-1} = i, q_t = j|m^l = \bar{m}, \lambda')
= \sum_{t=1}^{T} \sum_{\bar{m}=1}^{N} c'_m \xi_{i,j}^{l, \bar{m}}(t - 1), \quad (A.4)
\]

and further normalization over \(j\) gives (2.18). In addition, by the definition of \(\gamma_{i}^{l, \bar{m}}(t)\), it can be seen that \(P(O, q_t = i, m^l = \bar{m}|\lambda') = c'_m \gamma_{i}^{l, \bar{m}}(t)\). Then, (2.19) follows from (2.13).

A.3

Equations (2.20)–(2.22) will be derived here by applying approximations (A1)–(A3).

For \(\alpha\), it admits
\[ \begin{align*}
\text{def. of } \alpha_{i,j}^{t,m} (t + 1) &= P(O_1, \ldots, O_{t+1}, q_{t+1} = j | \lambda', m^l = m) \\
&= \sum_{i=0}^{S-1} P(O_i, \ldots, O_t, q_t = i, O_{t+1}, q_{t+1} = j | \lambda', m^l = m) \\
\text{def. of } \alpha \text{ and } (ii) &= \sum_{i=0}^{S-1} \alpha_{i,j}^{t,m} (t) P(O_{t+1} | q_{t+1} = j, O_1, \ldots, O_t, q_t = i, \lambda', m^l = m) P(q_{t+1} = j | q_t = i, \lambda', m^l = m) \\
&\approx \sum_{i=0}^{S-1} \alpha_{i,j}^{t,m} (t) P(O_{t+1} | q_{t+1} = j, q_t = i, \lambda', m^l = m) a_{i,j} \\
\text{def. of } \alpha \text{ and } (iv) &= \sum_{i=0}^{S-1} \alpha_{i,j}^{t,m} (t) P(o_{t+1}^l | q_{t+1} = j, \lambda', m^l = m) \cdot \prod_{r \neq t} P(o_{t+1}^r | q_{t+1} = j, \lambda') a_{i,j} \\
\text{def. of } b &= \sum_{i=0}^{S-1} \alpha_{i,j}^{t,m} (t) a_{i,j} b_j^{t,m} (o_{t+1}^l) \cdot \prod_{r \neq t} \left( \sum_{m=1}^{N} \delta_j^{t,m} (o_{t+1}^r) c_m \right) a_{i,j} \\
\text{def. of } \beta_{i,j}^{t,m} (t) &= \sum_{j=0}^{S-1} P(O_t, O_{t+1}, \ldots, O_T, q_{t+1} = j | q_t = i, \lambda', m^l = m) \\
\text{def. of } \beta \text{ and } (i) &= \sum_{j=0}^{S-1} a_{i,j} P(O_t, O_{t+1}, \ldots, O_T | q_{t+1} = j, q_t = i, \lambda', m^l = m) \\
\text{def. of } \beta \text{ and } (A2) &= \sum_{j=0}^{S-1} a_{i,j} P(O_{t+1}, \ldots, O_T | q_{t+1} = j, \lambda', m^l = m) \cdot P(O_t | q_t = i, \lambda', m^l = m) \\
\text{def. of } \beta &= \sum_{j=0}^{S-1} a_{i,j} \beta_{i,j}^{t,m} (t + 1) P(O_t | q_t = i, \lambda', m^l = m) \\
\text{def. of } b \text{ and } (iv) &= b_j^{t,m} (o_{t}^l) \prod_{r \neq t} \left( \sum_{m=1}^{N} \delta_j^{t,m} (o_{t}^r) c_m \right) \sum_{j=0}^{S-1} a_{i,j} \beta_{i,j}^{t,m} (t + 1). 
\end{align*} \]

For \( \beta \), it admits

\[ \sum_{j=0}^{S-1} P(O_t, O_{t+1}, \ldots, O_T, q_{t+1} = j | q_t = i, \lambda', m^l = m) \]

\[ \sum_{j=0}^{S-1} a_{i,j} P(O_t, O_{t+1}, \ldots, O_T | q_{t+1} = j, q_t = i, \lambda', m^l = m) \]

\[ \sum_{j=0}^{S-1} a_{i,j} P(O_{t+1}, \ldots, O_T | q_{t+1} = j, \lambda', m^l = m) \cdot P(O_t | q_t = i, \lambda', m^l = m) \]

\[ \sum_{j=0}^{S-1} a_{i,j} \beta_{i,j}^t (t + 1) P(O_t | q_t = i, \lambda', m^l = m) \]

\[ b_j^t (o_t^l) \prod_{r \neq t} \left( \sum_{m=1}^{N} \delta_j^{t,m} (o_t^r) c_m \right) \sum_{j=0}^{S-1} a_{i,j} \beta_{i,j}^t (t + 1). \]
For $\xi$, it admits

$$\xi_{ij}^{l,\bar{m}}(t) \triangleq P(O_t, q_t = i, q_{t+1} = j|\lambda', m^l = \bar{m})$$

$$= P(O_1, ..., O_t, q_t = i|\lambda', m^l = \bar{m})P(O_{t+1}, ..., O_T, q_{t+1} = j|O_1, ..., O_t, q_t = i, \lambda', m^l = \bar{m})$$

$$\approx \alpha_{i}^{l,\bar{m}}(t) P(q_{t+1} = j|q_t = i, \lambda')P(O_{t+1}, ..., O_T|q_{t+1} = j, \lambda', m^l = \bar{m})$$

$$\overset{\text{def. of } \beta}{=} \alpha_{i}^{l,\bar{m}}(t) a_{i,j}^{l,\bar{m}}(t + 1). \quad (A.7)$$

In the above, (i) the total probability theorem, (ii) the Bayes’ formula, (iii) independence between $O_{t+1}$ and $q_t$ given $q_{t+1}$ ($\forall t$), (iv) conditional independence of users’ observations given the true spectrum state are invoked.

A.4

In this appendix, we will show $SSA(B^H) \leq SSA(B^M)$. According to the definition of SSA and equations (3.1) and (3.2),

$$SSA(B^M) = |P_d^M - P_f^M|$$

$$= |(1 - \varphi_{10}) \cdot (P_d^H - P_f^H) - \varphi_{01} \cdot (P_d^H - P_f^H)|$$

$$= |1 - \varphi_{10} - \varphi_{01}| \cdot |P_d^H - P_f^H|$$

$$\leq 1 \cdot |P_d^H - P_f^H| = SSA(B^H), \quad (A.8)$$

where the fact $0 \leq \varphi_{10}, \varphi_{01} \leq 1$ is invoked.
B.1 Derivations of (3.3) and (3.4)

By the Bayesian formula and the total probability theorem, we have

\[
\Psi_1 \triangleq \Pr(r_t = 1 | r_{t-1} = 1) = \frac{\Pr(r_t = 1, r_{t-1} = 1)}{\Pr(r_{t-1} = 1)}
\]

(B.1)

\[
= \frac{1}{\sum_{i,j=0} \Pr(s_t = j, s_{t-1} = i) \Pr(r_t = 1, r_{t-1} = 1 | s_t = j, s_{t-1} = i)}
\]

\[
\times \sum_{i=0} \Pr(s_{t-1} = i) \Pr(r_{t-1} = 1 | s_{t-1} = i)
\]

\[
= \frac{\pi_0 a_{00} P^2_f + (\pi_0 a_{01} + \pi_1 a_{10}) P_f a + \pi_1 a_{11} P^2_d}{\pi_0 P_f a + \pi_1 P_d},
\]

where \( \Pr(r_t = 1, r_{t-1} = 1 | s_t = j, s_{t-1} = i) = \Pr(r_t = 1 | s_t = j) \Pr(r_{t-1} = 1 | s_{t-1} = i) \) is applied. Similar steps verify (3.4).

B.2 Proof of Proposition 4

Proof. It can be seen that \( \hat{\Psi}_1 = \frac{1}{n_1} \sum_{i=1}^{n_1} X_{t_i} \) in which \( X_{t_i} \) is defined as

\[
X_{t_i} \triangleq \begin{cases} 
1, & \text{if } r_{t_{i+1}} = 1, \text{ given } r_{t_i} = 1, \\
0, & \text{if } r_{t_{i+1}} = 0, \text{ given } r_{t_i} = 1,
\end{cases}
\]

(B.2)

where \( t_i \) is the time slot for the \( i \)-th reported 1 of the sensor, and \( n_1 \) is the number of pairs \( \{r_{t_i}, r_{t_{i+1}}\} \) with \( r_{t_i} = 1 \).\(^1\) To prove the convergence of \( \hat{\Psi}_1 \), we need to prove 1) \( E(\hat{\Psi}_1) = \Psi_1 \), which is simple to show by noticing that \( E(X_t) = \Pr(r_{t+1} = 1 | r_t = 1) = \Psi_1 \); 2) \( \lim_{T \to \infty} Var(\hat{\Psi}_1) = 0 \). In general, \( X_t \)'s are not independent due to the correlation between the consecutive true spectrum states in the Markov model. Thus, the central limit theorem can not be applied.

\(^1\)Similarly, \( n_0 \) is defined as the number of pairs \( \{r_{t_i}, r_{t_{i+1}}\} \) with \( r_{t_i} = 0 \).
However, we will show the second fact is true by first proving that the correlation between $X_i$ and $X_j$ ($i > j$) vanishes as $(i - j)$ approaches infinity. That is,

$$\lim_{(i - j) \to \infty} E(X_iX_j) = E(X_i)E(X_j).$$ \hfill (B.3)

Note that

$$E(X_iX_j)$$

$$= Pr(r_{i+1} = 1, r_{j+1} = 1 | r_i = 1, r_j = 1)$$

$$= Pr(r_{j+1} = 1 | r_j = 1) Pr(r_{i+1} = 1 | r_i = 1, r_j = 1)$$

$$= Pr(r_{j+1} = 1 | r_j = 1) [Pr(s_{i+1} = 1 | r_i = 1, r_j = 1) P_d + Pr(s_{i+1} = 0 | r_i = 1, r_j = 1) P_f],$$

and

$$E(X_j)E(X_i)$$

$$= Pr(r_{j+1} = 1 | r_j = 1) Pr(r_{i+1} = 1 | r_i = 1)$$

$$= Pr(r_{j+1} = 1 | r_j = 1) [Pr(s_{i+1} = 1 | r_i = 1) P_d + Pr(s_{i+1} = 0 | r_i = 1) P_f].$$

Comparing the two preceding equations, it can be seen that, to prove (B.3), it is sufficient to prove

$$\lim_{(i - j) \to \infty} Pr(s_{i+1} = 1 | r_i = 1, r_j = 1) = Pr(s_{i+1} = 1 | r_i = 1).$$

Note that $Pr(s_{i+1} = 1 | r_i = 1)$ is given by

$$Pr(s_{i+1} = 1 | r_i = 1)$$

$$= P_d^2 Pr(s_{i+1} = 1, s_i = 1) + P_f a Pr(s_{i+1} = 1, s_i = 0)$$

$$= \frac{P_d Pr(s_i = 1) + P_f a Pr(s_i = 0)}{\pi_1 P_d + \pi_0 P_f},$$

$$= \frac{\pi_1 P_d a_{11} + \pi_0 P_f a_{01}}{\pi_1 P_d + \pi_0 P_f},$$ \hfill (B.4)
and $Pr(s_{i+1} = 1|r_i = 1, r_j = 1)$ is given by

$$Pr(s_{i+1} = 1|r_i = 1, r_j = 1) = \frac{P_d^2 Pr(s_{i+1} = 1, s_i = 1, s_j = 1) + P_d P_f a Pr(s_{i+1} = 1, s_i = 0, s_j = 1)}{P_d^2 Pr(s_i = 1, s_j = 1) + P_d P_f a Pr(s_i = 0, s_j = 1) + P_d P_f a Pr(s_{i+1} = 1, s_i = 1, s_j = 0) + P_d^2 P_f a Pr(s_{i+1} = 1, s_i = 0, s_j = 0)}$$

... 

$$= \frac{\pi_1(P_d^2 p^{(1)}_{i-j} a_{11} + P_d P_f a (1 - p^{(1)}_{i-j}) a_{01}) + \pi_0(P_d^2 P_f a (1 - p^{(0)}_{i-j}) a_{01} + P_d P_f a (1 - p^{(0)}_{i-j}) a_{11})}{\pi_1(P_d^2 p^{(1)}_{i-j} + P_d P_f a (1 - p^{(1)}_{i-j})) + \pi_0(P_d^2 P_f a (1 - p^{(0)}_{i-j}) + P_d P_f a (1 - p^{(0)}_{i-j}))}, \quad (B.5)$$

where $p^{(1)}_n \triangleq Pr(s_{n+j} = 1|s_j = 1)$ and $p^{(0)}_n \triangleq Pr(s_{n+j} = 0|s_j = 0)$. According to the definition, the following recursive relation holds for $p^{(1)}_n$,

$$p^{(1)}_n = Pr(s_{j+n} = 1|s_j = 1) = Pr(s_{j+n} = 1, s_{j+n-1} = 1|s_j = 1) + Pr(s_{j+n} = 1, s_{j+n-1} = 0|s_j = 1) = a_{11} p^{(1)}_{n-1} + a_{01} (1 - p^{(1)}_{n-1}). \quad (B.6)$$

Consequently, $p^{(1)}_n = \frac{a_{01}}{1 - a_{11} + a_{01}}$. Similarly, we have $p^{(0)}_n = \frac{a_{10}}{1 - a_{01} + a_{10}}$. Substituting these two expressions into (B.5), it can be verified that $Pr(s_{i+1} = 1|r_i = 1, r_j = 1) = \frac{\pi_1 P_f a_{11} + \pi_0 P_f a_{01}}{\pi_1 P_f a_{11} + \pi_0 P_f a_{01}} = Pr(s_{i+1}|r_i = 1)$ as $i - j$ approaches infinity. Therefore (B.3) holds.

Now, we will use (B.3) to prove that $\lim_{n \to \infty} Var(\hat{\Psi}_1) = 0$. For any positive $\delta$, $\exists K_\delta$ such that $|Cov(X_i, X_j)| < \delta/2$ when $|i - j| > K_\delta$ due to (B.3). Also, given $K_\delta$, $\exists N_\delta$ such that $4K_\delta < \delta N_\delta$. Then, for any $n_1 > N_\delta$, we have $Var(\hat{\Psi}_1) = \frac{1}{n_1} \sum_i \sum_j Cov(X_i, X_j) \leq \frac{1}{n_1} \left[ n_1 \left( 1 \times 2 K_\delta + \frac{\delta}{2} \times (n_1 - 2K_\delta) \right) \right] < \frac{2K_\delta n_1}{n_1} K_\delta + \frac{\delta}{2} < \delta$. That is, $\lim_{n_1 \to \infty} Var(\hat{\Psi}_1) = 0$. On the other hand, for any finite $N_\delta$, $n_1 > N_\delta$ with probability 1 when $T$ approaches infinity, which implies $\lim_{T \to \infty} Var(\hat{\Psi}_1) = 0$. Therefore, $\hat{\Psi}_1$ converges to $\Psi_1$. Following the same approach, it can be shown that $\hat{\Psi}_0$ converges to $\Psi_0$. \hfill \Box

---

Note that $\ldots \frac{a+b}{a+b}$ is used to represent $\frac{a+b}{a+b}$ due to space limitations.
Appendix C

C.1 Proof of Proposition 7

Proof. The structure of this proof follows that of the Cheeger’s inequality given in [24].

1. For the LHS: First, it follows from (4.11) that
\[ \lambda_2(LW) \triangleq \inf_{g \perp \text{W}^{1/2}} \frac{g^T Lwg}{g^T g} = \inf_{f \perp \text{W}^1} \frac{f^T Lf}{f^T Wf}, \]  
with \( f = W^{-1/2}g \). Then choose \( \tilde{f} \) based on the optimal edge cut \( C \) which achieves \( h_W(L) \) and separates the vertices into two sets \( X \) and \( \bar{X} \):
\[ \tilde{f}(v) = \begin{cases} 1/|X|_W, & v \in X \\ -1/|\bar{X}|_W, & v \in \bar{X}. \end{cases} \]  
It can be verified that \( \tilde{f} \perp \text{W}^1 \), and thus we have
\[ \lambda_2(LW) \leq \frac{\tilde{f}^T L\tilde{f}}{f^T Wf} = \frac{\sum_{u \sim v} [\tilde{f}(u) - \tilde{f}(v)]^2 a_{u,v}}{\sum_v \tilde{f}^2(v) w_v} = \frac{1}{1/|X|_W + 1/|\bar{X}|_W} \sum_{u \in X, v \in X} a_{u,v} \leq \frac{2}{\min\{|X|_W, |\bar{X}|_W\}} = 2h_W(L), \]  
where \( u \sim v \) means that \( u \) and \( v \) are connected.

2. For the RHS: Assume that \( f \) achieves the infimum in (C.1), i.e., \( \lambda_2(LW) = \frac{f^T Lf}{f^T Wf} \), and then order the vertices such that
\[ f(v_1) \leq f(v_2) \leq \cdots \leq f(v_n). \]  

Without loss of generality, it is assumed that

\[
\sum_{f(v) \leq 0} w_v \geq \sum_{f(v) > 0} w_v, \tag{C.5}
\]

(otherwise, replace \( f \) by \(-f\)). Further define the \( i \)-cut as \( C_i = \{(v_p, v_q) \in E | 1 \leq p \leq i < q \leq n\} \) where \( E \) is the set of all edges, and denote

\[
\alpha_W = \min_{1 \leq i \leq n} \min \left\{ \frac{\sum_{(u,v) \in C_i} a_{u,v}}{\sum_{j \leq i} w_{v_j} + \sum_{j > i} w_{v_j}} \right\}, \tag{C.6}
\]

which admits \( \alpha_W \geq h_W(L) \) by definition. In addition, let \( V \) be the set of all nodes, \( V_+ = \{v | f(v) > 0\} \) and \( g(x) = \max\{0, f(x)\} \). By definition \( g(v_i) \) is increasing in \( i \), hence we have

\[
\sum_{u \sim v} a_{u,v}|g^2(u) - g^2(v)| = \sum_{i < j} a_{v_i,v_j}|g^2(v_i) - g^2(v_j)|
\]

\[
= \sum_{i < j} \left( a_{v_i,v_j} \sum_{k=i}^{j-1} \left( g^2(v_{k+1}) - g^2(v_k) \right) \right)
\]

\[
= \sum_{k=1}^{n-1} \left( g^2(v_{k+1}) - g^2(v_k) \right) \sum_{i \leq k < j} a_{v_i,v_j}. \tag{C.7}
\]

Since \( Lf = \lambda_2(L_W)Wf \), it follows that, for all vertices \( u \in V \),

\[
\sum_{\{v | u \sim v\}} [f(u) - f(v)]a_{u,v} = \lambda_2(L_W)f(u)w_u. \tag{C.8}
\]

Multiplying both sides of (C.8) with \( f(u) \) and further taking summation over all \( u \in V_+ \) leads to

\[
\sum_{u \in V_+} f(u) \sum_{\{v | u \sim v\}} [f(u) - f(v)]a_{u,v} = \lambda_2(L_W) \sum_{u \in V_+} f^2(u)w_u, \tag{C.9}
\]

and it follows that
\[ \lambda_2(L_W) = \frac{\sum_{u \in V_+} f(u) \sum_{v \in V_+} \sum_{u \sim v} [f(u) - f(v)] a_{u,v}}{\sum_{u \in V_+} f^2(u) w_u} \geq \frac{\sum_u g(u) \sum_{v \in V_+} [g(u) - g(v)] a_{u,v}}{\sum_u g^2(u) w_u} \]

\[ \geq \left( \sum_{u \sim v} [g(u) - g(v)]^2 a_{u,v} \right) \left( \sum_{u \sim v} [g(u) + g(v)]^2 a_{u,v} \right) \]

\[ \geq \left( \sum_{u \sim v} g^2(u) w_u \right) \left( \sum_{u \sim v} 2[g^2(u) + g^2(v)] a_{u,v} \right) \]

\[ \geq 2 \left( \sum_{u \sim v} g^2(u) w_u \right) \left( \sum_{u \sim v} g^2(u) \delta_u \right) \]

\[ \geq \frac{\alpha^2 W}{2} \cdot \frac{\sum_{i=1}^{n-1} \left( g^2(v_{i+1}) - g^2(v_i) \right) \sum_{(u,v) \in C_i} a_{u,v}}{\sum_u g^2(u) \delta_u} \geq \frac{\alpha^2 W}{2} \cdot \frac{\sum_{j>i} \left( g^2(v_{i+1}) - g^2(v_i) \right) a_W \sum_{j>i} w_{v_j}}{\sum_u g^2(u) \delta_u} \]

\[ = \frac{\alpha^2 W}{2} \cdot \frac{\sum_{i=1}^{n} g^2(v_i) w_{v_i}}{\sum_u g^2(u) \delta_u} \geq \frac{h_W^2(L)}{2} \cdot \frac{w_{\min}}{\delta_{\max}}, \tag{C.10} \]

where the first and the second inequalities are due to the definition of \( g(\cdot) \); the third inequality follows by the Cauchy-Schwarz inequality; the second equality follows from (C.7); the fourth inequality follows by (C.4), (C.5) and (C.6); the fifth inequality follows by changing of index and applying the fact \( g(v_1) = 0 \).

\[ \\Box \]

**C.2 Computing \( \frac{\partial a_{i,q}}{\partial x_i^{(q)}} \)**

First note that \( \frac{\partial a_{p,q}}{\partial x_i^{(k)}} = 0 \) when \( p = q \) and that \( \frac{\partial a_{p,q}}{\partial x_i^{(k)}} = \frac{\partial a_{q,p}}{\partial x_i^{(k)}} \) due to symmetry of \( a_{p,q} \). For \( k = 1, 2 \), we have

\[ 177 \]
\[ \frac{\partial a_{p,q}}{\partial x_i^{(k)}} = B \cdot \left( \frac{1}{\ln(1 + \text{SIR}_{p,q})} + \frac{1}{\ln(1 + \text{SIR}_{q,p})} \right)^{-2} \left[ \frac{1}{(1 + \text{SIR}_{p,q}) \ln^2(1 + \text{SIR}_{p,q})} \frac{\partial \text{SIR}_{p,q}}{\partial x_i^{(k)}} \right. \]
\[ + \left. \frac{1}{(1 + \text{SIR}_{q,p}) \ln^2(1 + \text{SIR}_{q,p})} \frac{\partial \text{SIR}_{q,p}}{\partial x_i^{(k)}} \right], \]  
(C.11)

where \( \frac{\partial \text{SIR}_{p,q}}{\partial x_i^{(k)}} \) and \( \frac{\partial \text{SIR}_{q,p}}{\partial x_i^{(k)}} \) are given below.

When \( p = i \) and \( p \neq q \),
\[ \frac{\partial \text{SIR}_{i,q}}{\partial x_i^{(k)}} = \frac{1}{\eta \sum_{j=1}^{l} d_{i,j}^{-\alpha} + \sum_{s \in \Xi, i,q} u(d_{q,s}/r_{int})} \left[ \alpha \cdot d_{i,q}^{-\alpha - 2} \cdot \left( x_q^{(k)} - x_i^{(k)} \right) \right], \]  
(C.12)

and
\[ \frac{\partial \text{SIR}_{p,i}}{\partial x_i^{(k)}} = \frac{1}{\eta \sum_{j=1}^{l} d_{i,j}^{-\alpha} + \sum_{s \in \Xi, i,q} u(d_{i,s}/r_{int})} \left[ \alpha \cdot d_{i,q}^{-\alpha - 2} \cdot \left( x_i^{(k)} - x_i^{(k)} \right) + \sum_{s \in \Xi, i,q} u(d_{i,s}/r_{int}) \frac{x_s^{(k)} - x_i^{(k)}}{r_{int} \cdot d_{i,s}} \right], \]  
(C.13)

respectively. In (C.13), \( \dot{u}(x) \triangleq \frac{du}{dx} \) is the derivative of \( u \) with respect to \( x \).

When \( p, q \neq i \) and \( p \neq q \),
\[ \frac{\partial \text{SIR}_{p,q}}{\partial x_i^{(k)}} = -\frac{d_{p,q}^{-\alpha}}{\eta \sum_{j=1}^{l} d_{j,q}^{-\alpha} + \sum_{s \in \Xi, p,q} u(d_{q,s}/r_{int})} \cdot \dot{u}(d_{q,i}/r_{int}) \cdot \frac{1}{r_{int}} \cdot \frac{x_i^{(k)} - x_q^{(k)}}{d_{q,i}} , \]  
(C.14)

and \( \frac{\partial \text{SIR}_{q,p}}{\partial x_i^{(k)}} \) can be obtained similarly by switching the subscripts \( p \) and \( q \) in (C.14).

### C.3 Proof of Proposition 8

**Proof.** 1. For the LHS: Choose \( f \) based on the optimal edge cut \( C \) which achieves \( h_{\text{W}}^{(M)}(L) \) and separates the vertices into two sets \( X \) and \( \bar{X} \):

178
\[ f(v) = \begin{cases} 
1/|X|, & v \in X \\
-1/|\bar{X}|, & v \in \bar{X}.
\end{cases} \quad (C.15) \]

It can be verified that \( f \perp \hat{W}1 \), and thus we further have (for \( 0 \leq k < 1 \))

\[
\lambda_2(L^{(M)}_W) \leq \frac{f^T L f}{f^T \hat{W} f} = \sum_{u,v} \frac{|f(u) - f(v)|^2 a_{u,v}}{\sum_v f^2(v) w_v} \\
\frac{1/|X| + 1/|\bar{X}|}{1/|X| + 1/|\bar{X}|} \leq 2 \sum_{u,v} a_{u,v} \\
\min \left\{ \sum_{m=1}^M \left( \sum_{v_i \in X} w_i^{(m)} \right)^{1-k}, \sum_{m=1}^M \left( \sum_{v_i \in \bar{X}} w_i^{(m)} \right)^{1-k} \right\} \\
\leq 2 \sum_{u,v} a_{u,v} \\
\min \left\{ \sum_{m=1}^M \left( \sum_{v_i \in X} w_i^{(m)} \right)^{1-k}, \sum_{m=1}^M \left( \sum_{v_i \in \bar{X}} w_i^{(m)} \right)^{1-k} \right\} \\
\leq 2 \sum_{m=1}^M \min \left\{ \left( \sum_{v_i \in X} w_i^{(m)} \right)^{1-k}, \left( \sum_{v_i \in \bar{X}} w_i^{(m)} \right)^{1-k} \right\} \\
= 2 \sum_{m=1}^M \min \left\{ |X|^{1-k}_{\hat{W}(m)}, |\bar{X}|^{1-k}_{\hat{W}(m)} \right\} \\
= 2 \sum_{m=1}^M \min \left\{ |X|^{1-k}_{\hat{W}(m)}, |\bar{X}|^{1-k}_{\hat{W}(m)} \right\} \\
= 2 \sum_{m=1}^M \min \left\{ |X|^{1-k}_{\hat{W}(m)}, |\bar{X}|^{1-k}_{\hat{W}(m)} \right\} = 2h^{(M)}_{\hat{W}}(L), \quad (C.16) \]

where \( \sum_{v_i} w_i^{(m)} \geq \left( \sum_{v_i} w_i^{(m)} \right)^{1-k} \) (due to concavity of the function \( x^{1-k} \) and Jensen’s inequality) is used in the third inequality, and the fact \( \min\{\sum a_i, \sum b_i\} \geq \sum \min\{a_i, b_i\} \) is invoked in the fourth inequality.

2. For the RHS: For any \( S \subset V \), first notice that
\[
\min \{ |S|_W, |\bar{S}|_W \} \\
= \min \left\{ \sum_{m=1}^{M} \sum_{v_i \in S} (w_i^{(m)})^{1-k}, \sum_{m=1}^{M} \sum_{v_i \in \bar{S}} (w_i^{(m)})^{1-k} \right\} \\
\leq n^k \cdot \min \left\{ \sum_{m=1}^{M} \left( \sum_{v_i \in S} (w_i^{(m)})^{1-k} \right), \sum_{m=1}^{M} \left( \sum_{v_i \in \bar{S}} (w_i^{(m)})^{1-k} \right) \right\} \\
\leq n^k \cdot \xi(W) \sum_{m=1}^{M} \min \left\{ \left( \sum_{v_i \in S} (w_i^{(m)})^{1-k} \right), \left( \sum_{v_i \in \bar{S}} (w_i^{(m)})^{1-k} \right) \right\} \\
= n^k \cdot \xi(W) \sum_{m=1}^{M} \min^{1-k} \{ |S|_{W(m)}, |\bar{S}|_{W(m)} \}, \quad (C.17)
\]

where \( \sum_{v_i} (w_i^{(m)})^{1-k} \leq n^k \cdot \left( \sum_{v_i} w_i^{(m)} \right)^{1-k} \) is applied in the first inequality, which in turn is obtained through the Hölder’s inequality:

\[
\sum_i |x_i y_i| \leq \left( \sum_i |x_i|^p \right)^{1/p} \cdot \left( \sum_i |y_i|^q \right)^{1/q}, \quad (C.18)
\]

with \( x_i = (w_i^{(m)})^{1-k}, y_i = 1, p = \frac{1}{1-k} \) and \( q = \frac{1}{k} \). Defining \( h_W(L) \) by replacing \( W \) with \( \tilde{W} \) in (4.10), it can be inferred from (4.17) and (C.17) that

\[
h_{\tilde{W}}(L) \triangleq \min_{S} \sum_{i \in S, j \in \bar{S}} a_{i,j} \\
\geq \min_{S} \frac{1}{n^k \cdot \xi(W) \sum_{m=1}^{M} \min^{1-k} \{ |S|_{W(m)}, |\bar{S}|_{W(m)} \}} \\
\sum_{i \in S, j \in \bar{S}} a_{i,j} \\
= h_{\tilde{W}}^{(M)}(L) = \frac{h_{\tilde{W}}^{(M)}(L)}{n^k \xi(W)}. \quad (C.19)
\]

Further, replacing \( W \) by \( \tilde{W} \) in (C.10), we have
\[
\lambda_2(L^{(M)}_{\tilde{W}}) \geq \frac{h_{\tilde{W}}^2(L)}{2\delta_{max}} \cdot \tilde{w}_{min} \geq \frac{(h^{(M)}_W(L))^2}{2\delta_{max}} \cdot \frac{\tilde{w}_{min}}{\eta^{2k}(\xi^{(M)}_W)^2}.
\] (C.20)
Appendix D

D.1 Proof of (5.11)

Since $Q^{(mp)}_s$ is identical to $Q^{(m)}_s$ (c.f. (6.2)) by definition, it admits the following expansion

$$Q^{(mp)}_s(s, a, o) \triangleq E_{R,S'}[R(s, a, o) + \beta V^{(m)}_s(S')]$$

$$= E_S[R(s, a, o)] + \beta \sum_{s'} p(s'|s, a, o)V^{(mp)}_s(s')$$

$$= r^k(s, a, o) + \sum_{\tilde{s}} p^k(\tilde{s}|s, a) \left[ r^u(\tilde{s}, a, o) + \beta \sum_{s'} p^u(s'|\tilde{s}, a, o)V^{(mp)}_s(s') \right]$$

$$= r^k(s, a, o) + \sum_{\tilde{s}} p^k(\tilde{s}|s, a)\tilde{Q}^{(mp)}_s(\tilde{s}, a, o), \quad (D.1)$$

where the second equality invokes the facts that the randomness of $R(s, a, o)$ is solely due to $\tilde{S}$ and that $V^{(mp)}_s$ is the same as $V^{(m)}_s$ by definition; the third equality follows from (5.8) and (5.9); the last one is by the definition of $\tilde{Q}^{(mp)}_s$ in (5.10).

D.2 Proof of Proposition 9

To streamline the proof for Propositions 9 (as well as Proposition 10), several relevant definitions and useful lemmas are introduced first.

**Definition 5.** The max-norm of an $n$-dimensional matrix $A = [a_{i_1,\ldots,i_n}]$ is defined as $\|A\|_\infty = \max_{i_1,\ldots,i_n} |a_{i_1,\ldots,i_n}|$, and unless otherwise noted, $\|\cdot\|_\infty$ represents the max-norm.

**Definition 6.** For any two $m \times n$ matrices $A = [a_{i,j}]$ and $B = [b_{i,j}]$, $A \geq B$ if $a_{i,j} \geq b_{i,j}$ for $i = 1,\ldots,m$ and $j = 1,\ldots,n$.

**Lemma 1.** (Szepesvari and Littman [185]) Assume a learning rate sequence $\alpha_n$ that satisfies $0 \leq \alpha_n < 1$, $\sum_{n=0}^\infty \alpha_n = \infty$ and $\sum_{n=0}^\infty \alpha_n^2 < \infty$, and a sequence of (random) mappings $T_n$ from
Particularly, for any quality function $Q$ and $V$-matrix for state $|A| \times |O|$

For any two quality functions ($Q_1$ and $Q_2$) and corresponding value functions ($V_1$ and $V_2$),

$$|V_1(s) - V_2(s)| \leq ||Q_1(s) - Q_2(s)||_\infty.$$  \hfill (D.3)

Particularly, for any quality function $Q$ in minimax-PDS, $Q(s) \triangleq [Q(s, a, o)]_{a \in A, o \in O}$ is the $|A| \times |O|$ Q-matrix for state $s$, with $|A|$ and $|O|$ the cardinalities of action sets of the LS and the attacker, respectively; in WoLF-PDS, $Q(s) \triangleq [Q(s, a)]_{a \in A}$ reduces to a $|A| \times 1$ vector.

**Proof.** The minimax-PDS case is proved first. Denoting $1_{|A|}$ and $1_{|O|}$ all-one column vectors of length $|A|$ and $|O|$, respectively, it can be noticed that for all $s$,

$$Q_2(s) \geq Q_1(s) - ||Q_1(s) - Q_2(s)||_\infty 1_{|A|}^T 1_{|O|},$$

$$Q_2(s) \leq Q_1(s) + ||Q_2(s) - Q_1(s)||_\infty 1_{|A|}^T 1_{|O|}. \hfill (D.4)$$

Then applying the fact that $\text{val}(Q'(s)) \geq \text{val}(Q''(s))$ for any matrices $Q'(s) \geq Q''(s)$ (where $\text{val}(Q(s)) \triangleq \max_{\pi(s)} \min_o \sum_a Q(s, a, o)\pi_a(s)$ in minimax-PDS) on (D.4), we have

$$V_2(s) = \text{val}(Q_2(s))$$

$$\geq \text{val} \left( Q_1(s) - ||Q_1(s) - Q_2(s)||_\infty 1_{|A|}^T 1_{|O|} \right)$$

$$= V_1(s) - ||Q_1(s) - Q_2(s)||_\infty, \hfill (D.5)$$

and

$$V_2(s) = \text{val}(Q_2(s))$$

$$\leq \text{val} \left( Q_1(s) + ||Q_2(s) - Q_1(s)||_\infty 1_{|A|}^T 1_{|O|} \right)$$

$$= V_1(s) + ||Q_2(s) - Q_1(s)||_\infty, \hfill (D.6)$$

which implies $|V_1(s) - V_2(s)| \leq ||Q_1(s) - Q_2(s)||_\infty$. 

183
For the WoLF-PDS case, we have

\[ |V_1(s) - V_2(s)| = |\max_a Q_1(s, a) - \max_a Q_2(s, a)| \leq \max_a |Q_1(s, a) - Q_2(s, a)| = ||Q_1(s) - Q_2(s)||_\infty. \]  

(D.7)

Now we are ready to prove Proposition 9.

Proof. For simplicity, the superscript \((mp)\) will be omitted in this proof. To show \(Q_n\) converges to \(Q^*\) in Proposition 9, it is sufficient to show \(\tilde{Q}_n\) converges to \(\tilde{Q}^*\), considering the definitions of \(Q_n\) and \(Q^*\) in minimax-PDS (c.f. (5.11)). To this end, let the mapping sequence \(T_n\) be such that

\[ (T_n\tilde{Q})(\tilde{s}_n, a_n, o_n) \triangleq r^u(\tilde{s}_n, a_n, o_n) + \beta \cdot V(s_{n+1}), \]  

(D.8)

and for \((\tilde{s}, a, o) \neq (\tilde{s}_n, a_n, o_n)\),

\[ (T_n\tilde{Q})(\tilde{s}, a, o) \triangleq \tilde{Q}(\tilde{s}, a, o). \]  

(D.9)

First notice that (C1) in Lemma 1 holds, since

\[
\mathbb{E}_{S_{n+1}}[(T_n\tilde{Q}^*)(\tilde{s}_n, a_n, o_n)] \\
= \sum_{s_{n+1}} p^u(s_{n+1}|\tilde{s}_n, a_n, o_n)[r^u(\tilde{s}_n, a_n, o_n) + \beta \cdot V_*(s_{n+1})] \\
= r^u(\tilde{s}_n, a_n, o_n) + \beta \sum_{s_{n+1}} p(s_{n+1}|\tilde{s}_n, a_n, o_n)V_*(s_{n+1}) \\
= \tilde{Q}^*(\tilde{s}_n, a_n, o_n),
\]

where the last step follows from the definition in (5.10), and \(\mathbb{E}[(T_n\tilde{Q}^*)(\tilde{s}, a, o)] = \tilde{Q}^*(\tilde{s}, a, o)\) also
holds for \((\tilde{s}, a, o) \neq (s_n, a_n, o_n)\). The contraction property (C2) is shown as follows for any \(\tilde{Q}\):

\[
\|T_n \tilde{Q} - T_n \tilde{Q}_*\|_\infty \\
\leq \beta \cdot \max_{s'} |V(s') - V_*(s')| \\
\leq \beta \cdot \max_{s'} ||Q(s') - Q_*(s')||_\infty \\
= \beta \cdot \max_{s'} \max_{a, o} |\sum \tilde{s} p^k(\tilde{s}|s', a, o) (\tilde{Q}(\tilde{s}, a, o) - \tilde{Q}_*(\tilde{s}, a, o))| \\
\leq \beta \cdot \max_{s'} \max_{a, o} \sum \tilde{s} p^k(\tilde{s}|s', a, o) |\tilde{Q}(\tilde{s}, a, o) - \tilde{Q}_*(\tilde{s}, a, o)| \\
\leq \beta \cdot \max_{a, o} \max_{s} |\tilde{Q}(\tilde{s}, a, o) - \tilde{Q}_*(\tilde{s}, a, o)| \\
= \beta \cdot ||\tilde{Q} - \tilde{Q}_*||_\infty,
\]

where the first inequality is due to (D.8); the second inequality is due to Lemma 2; and the third inequality is due to triangle inequality. Hence by Lemma 1 with \(\lambda_n = 0\), \(\tilde{Q}\) converges to \(\tilde{Q}_*\) with probability 1 for learning rate sequence \(\alpha_n\) satisfying the conditions there. \(\square\)

### D.3 Proof of (5.14)

As \(Q_*(wp)\) is defined identically to \(Q_*(w)\) (c.f. (5.5)), it admits the following expansion

\[
Q_*(wp)(s, a) \\
= E_{O, R, S'} \left[ R(s, a, O) + \beta \cdot V_*(w)(S') \right] \\
= E_{O, S}[R(s, a, O)] + E_O \left[ \beta \sum \tilde{s} p(s'|s, a, o) \cdot V_*(wp)(s') \right] \\
= E_O \left[ r^k(s, a, O) \right] + E_O \left[ \sum \tilde{s} p^k(\tilde{s}|s, a, O) r^u(s, a, O) \right] \\
+ E_O \left[ \beta \sum \tilde{s'} \sum \tilde{s} p^u(s'|\tilde{s}, a, O) p^k(\tilde{s}|s, a, O) V_*(wp)(s') \right] \\
= E_O [r^k(s, a, O)] + \sum \tilde{s} p^k(\tilde{s}|s, a) \tilde{Q}_*(wp)(\tilde{s}, a),
\]

where the third equality is due to assumptions (5.8) and (5.9) and the fact that \(V_*(wp)\) is the same as \(V_*(w)\) by definition; the last one follows from (5.13) and (5.15).
D.4 Proof of Proposition 10

Proof. For simplicity, the superscript \((wp)\) will be omitted in this proof. First notice that in WoLF [108], the learnt policy \(\pi_n\) will converge to the best response as long as the corresponding \(Q_n\) converges to \(Q^*\), when facing a stationary opponent. Further notice that WoLF-PDS and WoLF admit the same relationship between \(\pi_n\) and \(Q_n\). Therefore, to show the rationality of the proposed WoLF-PDS, it is sufficient to show \(Q_n\) defined in WoLF-PDS converges to \(Q^*\). For this purpose, we will adopt a similar approach as in the proof of Proposition 9. The convergence of \(\tilde{Q}\) will be shown first and then the convergence of \(Q\) follows from the definitions of \(Q_n\) and \(Q^*\) in WoLF-PDS (c.f. (5.14)). Define the mapping sequence \(T_n\) as

\[
(T_n\tilde{Q})(\tilde{s}_n, a_n) \triangleq r^u(\tilde{s}_n, a_n, o_n) + \beta \cdot V(s_{n+1}), \tag{D.13}
\]

and for \((\tilde{s}, a) \neq (\tilde{s}_n, a_n),\)

\[
(T_n\tilde{Q})(\tilde{s}, a) \triangleq \tilde{Q}(\tilde{s}, a). \tag{D.14}
\]

It follows from (5.14) that for a stationary opponent, (C1) in Lemma 1 holds since

\[
\begin{align*}
\mathbb{E}_{O_n, S_{n+1}}[T_n\tilde{Q}^* (\tilde{s}_n, a_n)] \\
= \mathbb{E}_{O_n} \left[ \sum_{s_{n+1}} p^u(s_{n+1} | \tilde{s}_n, a_n, O_n) \left( r^u(\tilde{s}_n, a_n, O_n) + \beta \cdot V(s_{n+1}) \right) \right], \\
= \mathbb{E}_{O_n} \left[ r^u(\tilde{s}_n, a_n, O_n) + \beta \sum_{s_{n+1}} p^u(s_{n+1} | \tilde{s}_n, a_n, O_n) V(s_{n+1}) \right] \\
= \tilde{Q}^*(\tilde{s}_n, a_n),
\end{align*}
\]

and \(\mathbb{E}[(T_n\tilde{Q}^*)(\tilde{s}, a)] = \tilde{Q}^*(\tilde{s}, a)\) also holds for \((\tilde{s}, a) \neq (\tilde{s}_n, a_n)\). Further notice that

\[
\begin{align*}
||T_n\tilde{Q} - T_n\tilde{Q}^*||_\infty \\
\leq \beta \cdot \max_{s' } |V(s') - V^*(s')| \leq \beta \cdot \max_{s' } ||Q(s') - Q^*(s')||_\infty \\
= \beta \cdot \max_{s', a, o} \left| \sum_{\tilde{s}} p^k(\tilde{s}|s', a) \left( \tilde{Q}(\tilde{s}, a, o) - \tilde{Q}^*(\tilde{s}, a, o) \right) \right| + \lambda_n \\
\leq \beta \cdot \max_{s', a, o} \left| \sum_{\tilde{s}} p^k(\tilde{s}|s', a) \left( \tilde{Q}(\tilde{s}, a, o) - \tilde{Q}^*(\tilde{s}, a, o) \right) \right| + \lambda_n \\
\leq \beta \cdot \max_{s', a, o} \left| \tilde{Q}(\tilde{s}, a, o) - \tilde{Q}^*(\tilde{s}, a, o) \right| + \lambda_n \\
= \beta \cdot ||\tilde{Q} - \tilde{Q}^*||_\infty + \lambda_n, \tag{D.16}
\end{align*}
\]
where similar arguments as in the proof of Proposition 9 are invoked for the above derivation, and \( \lambda_n \) is defined as

\[
\lambda_n \triangleq \max_{s,a} |r^k_n(s,a) - \mathbb{E}_O[r^k(s,a,O)]|. \tag{D.17}
\]

Hence, if \( \lambda_n \) converges to zero with probability 1, it follows that (C2) in Lemma 1 holds. This implies that, with suitable learning rate, \( \bar{Q}_n \) converges to \( \bar{Q}_* \) with probability 1.

To show that \( \lambda_n \) converges to zero with probability 1, it is sufficient to show \( \bar{r}_n^k(s,a) \xrightarrow{w.p.} \mathbb{E}_O[r^k(s,a,O)] \) for all \((s,a)\). To this end, for each \((s,a)\), consider a mapping sequence \( F_n \) such that, for any real number \( r \in \mathbb{R} \),

\[
F_n r \triangleq r^k(s,a,o_n), \tag{D.18}
\]

where the image \( r^k(s,a,o_n) \) of the mapping \( F_n \) is independent of its pre-image \( r \). Further notice that (5.17) can be written as

\[
\bar{r}_{n+1}^k(s,a) = (1 - \alpha_n) \bar{r}_n^k(s,a) + \alpha_n \cdot F_n \bar{r}_n^k(s,a). \tag{D.19}
\]

Then, it follows from the definition in (D.18) and the stationarity of the opponent’s policy that

\[
\mathbb{E}[F_n \mathbb{E}_O[r^k(s,a,O)]] = \mathbb{E}_O[r^k(s,a,O)], \tag{D.20}
\]

and that for any \( r \in \mathbb{R} \) and \( \gamma \in (0,1) \),

\[
|F_n r - F_n \mathbb{E}_O[r^k(s,a,O)]| \leq \gamma \cdot |r - \mathbb{E}_O[r^k(s,a,O)]|. \tag{D.21}
\]

These imply that (C1) and (C2) of Lemma 1 hold for \( F_n \) (by reducing \( Q \) and \( \bar{Q} \) to a \(|S|\)-dimensional vector and \( \mathbb{R}^{|S|} \), respectively, with \(|S|\) the cardinality of the state space), and hence \( \bar{r}_n^k(s,a) \xrightarrow{w.p.} \mathbb{E}_O[r^k(s,a,O)] \).

\[\qed\]

\[1\]Note that in the special case of \( \alpha_n = \frac{1}{n} \), \( \bar{r}_n^k(s,a) \xrightarrow{w.p.} \mathbb{E}_O[r^k(s,a,O)] \) readily follows from the well-known weak law of large numbers.
Appendix E

E.1 Proof of Proposition 11

Sketch of proof. With suitable learning rate and exploration, it follows from [107] and [128] that $Q_n^{(e)}$ and $Q_n^{(f)}$ converge to $Q^*_e$ and $Q^*_f$, respectively. Moreover, the assumption of identical discounting factor $\beta$ ensures that, the quality function of user-II will also converge to either $Q^*_e$ or $Q^*_f$, depending on its true type. Therefore, when user-II is truly an enemy, the strategy $\pi^H_n$ learned by user-II itself and the strategy $\hat{\pi}_n^{H(e)}$ estimated by user-I will converge to the same value $\pi^H_e$; similar conclusions can be drawn for the case of friend-type user-II as well.

Once both users learn the above mentioned optimal strategies, the belief $b_n$ will gradually converge to the true value. To see this, first notice that the optimal strategy of user-II can be expressed as a probability function $P_{b_n^{(f)}} = b^{(f)}_n ([1 - p_{explr}) \cdot \pi^H(e) + p_{explr}/m] + (1 - b^{(f)}_n) \cdot [(1 - p_{explr}) \cdot \pi^H(f) + p_{explr}/m]$ parameterized by $b^{(f)}_n$, with $b^{(f)}_n = 1$ when the true type of user-II is friend and $b^{(f)}_n = 0$ otherwise; further notice that user-II’s actions will be drawn repeatedly according to $P_{b_n^{(f)}}$ after the convergence (say, at timeslot $n = i_0$) of user-II’s strategy. Then, it is not difficult to see that $b^{(f)}_n$ (for $n > i_0$) is the posterior probability that user-II is friend-type (with a prior $b^{(f)}_{i_0}$), given that the same $p_{explr}$ is used by both users. Therefore, Proposition 11 follows from the result in [186] which states that for any prior, the posterior will converge almost surely to the true distribution. \qed
Appendix F

F.1 Proof of Proposition 1

Proof. Let $S = A - B^T C^{-1} B$ be the Schur complement of block $C$ in $\Gamma$. Then,

$$MSE(\hat{h}_{T_l,R_l}) = \mathbb{E}[(\hat{h}_{T_l,R_l} - h_{T_l,R_l})^2] = A - B^T C^{-1} B$$

$$= \det(A - B^T C^{-1} B) \det(S) = \frac{\det(\Gamma)}{\det(C)} \quad (F.1)$$

\[ \square \]

F.2 Proof of Proposition 2

Proof. It is assumed without loss of generality that the first $k$ adversary channels $\{h_{T_l,R_{a_1}}, \ldots, h_{T_l,R_{a_k}}\}$ are used. For clarity, let $x = h_{T_l,R_l} - \mathbb{E}[h_{T_l,R_l}]$, $y_i = h_{T_l,R_{a_i}} - \mathbb{E}[h_{T_l,R_{a_i}}]$, and $y^{(m)} = [y_1, y_2, \ldots, y_m]^T$ ($m \leq n$). Consequently, $\mathbb{E}[x] = 0$, and $\mathbb{E}[y_i] = 0$ ($i = 1, \ldots, n$). It is clear that, the covariance matrices corresponding to $h_{T_l,R_l}$ and $\{h_{T_l,R_{a_i}}\}_{i=1}^m$ are identical to those corresponding to $x$ and $y^{(m)}$. Consequently, the coefficient vector in the estimator of $x$ based on measurement $y^{(n)}$ is also $B^T C^{-1}$ (c.f. (8.5)) and will be denoted by $\xi$ in the following. Define $C_m \triangleq Cov(y^{(m)}, y^{(m)})$ and $B_m \triangleq Cov(x, y^{(m)})$ for all $m \leq n$. Then, $\xi = \xi_n$, where $\xi_m \triangleq C_m^{-1} B_m$ for all $m \leq n$. Further, it can be verified that $MSE(\hat{x}_m)$, the MSE of the estimate of $x$ based on $y^{(m)}$, is the same as the MSE of the legitimate channel estimate based on $\{h_{T_l,R_{a_1}}, \ldots, h_{T_l,R_{a_m}}\}$.

Proposition 12 holds if $MSE(\hat{x}_{m+1}) \leq MSE(\hat{x}_m)$ for all $m \leq n - 1$ by induction. Let $D_{m+1} \triangleq Cov(y_{m+1}, y^{(m)})$ and $c_{m+1} \triangleq Cov(y_{m+1}, y_{m+1})$. It is not difficult to see the following facts:

1. $MSE(\hat{x}_m) = \mathbb{E}[x^2] - B_m^T C_m^{-1} B_m$,

2. $C_{m+1} = \begin{bmatrix} C_m & D_{m+1} \\ D_m^T & c_{m+1} \end{bmatrix}$,
3. \([C_m \ D_{m+1}]\xi_{m+1} = B_m\) (due to \([C_{m+1}]\xi_{m+1} = B_{m+1}\)).

Decomposing \(\xi_{m+1}\) as \(\xi_{m+1} = [\alpha_{m+1}^T \ \beta]^T\), the above facts lead to the following equivalence relations:

\[
MSE(\hat{x}_{m+1}) \leq MSE(\hat{x}_m) \\
<= \beta^2 c_{m+1} \geq \beta^2 D_{m+1}^T C_m^{-1} D_{m+1}. \quad (F.2)
\]

If \(\beta = 0\), the proof is completed. Otherwise, it remains to show that \(c_{m+1} - D_{m+1}^T C_m^{-1} D_{m+1} \geq 0\).

Note that the left hand side of this inequality is the Schur complement of \(C_m\) in \(C_{m+1}\), denoted as \(S(C_m)\). Further applying the facts that \(\det(C_m) > 0\) and \(\det(C_{m+1}) \geq 0\), it leads to \(c_{m+1} - D_{m+1}^T C_m^{-1} D_{m+1} = \det(S(C_m)) = \frac{\det(C_{m+1})}{\det(C_m)} \geq 0.1\)

\[\square\]

F.3 Proof of Corollary 1

**Proof.** With the given assumptions, \(\Gamma\) (defined in Proposition 13) is of the form

\[
\Gamma = \begin{bmatrix}
1 & \rho & \cdots & \rho \\
\rho & 1 & \cdots & \rho' \\
\rho & \rho' & 1 & \cdots \\
\vdots & \vdots & \ddots & \vdots \\
\rho & \rho' & \cdots & 1
\end{bmatrix}_{(n+1) \times (n+1)}.
\]

Due to the circulant structure of \(\Gamma\), it can be shown that

\[
\det(\Gamma) = \prod_{i=0}^{n-1} \left( c_0 + c_1 \omega_1^i + \cdots + c_{n-1} \omega_n^{(n-1)} \right), \quad (F.3)
\]

\[\text{Here, a non-singular } C_m \text{ is assumed because the adversary channels that are linear combinations of the others can always be discarded.}\]
where $\omega_k = \exp(j \frac{2\pi k}{n})$ is the $n$th roots of unity, and $c_0 = 1 - \rho^2$, $c_i = \rho' - \rho^2$ for $i > 0$. Similarly, the determinant of $C$ is given by

$$\det(C) = \prod_{i=0}^{n-1} \left( c'_0 + c'_1 \omega_i^1 + \cdots + c'_{n-1} \omega_i^{n-1} \right), \quad (F.4)$$

where $c'_0 = 1$ and $c'_i = \rho'$ for $i > 0$. Then, according to Proposition 13, the MSE of the estimator $\hat{h}_{T_1, R_i}$ based on the channel measurements from these $n$ adversary receivers is given by

$$MSE(\hat{h}_{T_1, R_i}) = \det(C) \left( \sum_{k=0}^{n-1} \omega_k \right)$$

where in the second last step the fact $\sum_{k=0}^{n-1} \omega_k = \delta(i)$ is applied. Then, Corollary 1 follows readily by setting $MSE(\hat{h}_{T_1, R_i}) = 0$ in (F.5). \hfill \square