

## ABSTRACT

SHEN, WENBO. Physical Layer Signal Design and Control For Wireless Security. (Under the direction of Dr. Peng Ning and Dr. Huaiyu Dai.)

In recent decades, wireless communication technology has been widely deployed and increasingly adopted. As wireless devices become ubiquitous, more and more attacks targeting wireless devices emerge, including the wireless jamming attack and the man-in-the-middle attack to the wireless pairing process. On the other hand, the recent advances in the software-defined radio give us the flexibility to design and control wireless physical layer signals. We found that the wireless physical layer signal has some properties, which can be used to preserve wireless security.

In this dissertation, we first proposed a novel technique - *Multi-Channel Ratio (MCR) Decoding*, which aims at providing the anti-jamming wireless communication capability for multi-antenna wireless devices. The basic idea of MCR decoding is to fully leverage the repeated preamble signals and the multi-channel characteristics in MIMO communications to detect and recover desired transmission signals under constant and reactive jamming attacks. It is shown that the proposed MCR decoding can detect the desired transmission reliably under the jamming attack and remove jamming signals effectively.

After that, we explored the feasibility of using jamming to achieve wireless access control. We proposed a novel mechanism, called *Ally Friendly Jamming*, which provides an intelligent jamming capability that can disable unauthorized wireless communication but at the same time still allow authorized wireless devices to communicate, even if all these devices operate at the same frequency. The basic idea is to jam the wireless channel continuously but properly control the jamming signals with secret keys, so that the jamming signals are unpredictable interference to unauthorized devices, but are removable at authorized ones equipped with the secret keys. To achieve ally friendly jamming, we developed new techniques to generate ally jamming signals, to remove jamming signals from multiple ally jammers. Both the analytical and experimental results indicated that the proposed techniques can effectively disable enemy wireless communication and at the same time maintain wireless communication between authorized devices.

We further proposed *Fast Friendly Jamming*, which eliminates the need for demodulation and enables the friendly jammer to verify the received signals directly on the physical layer. We have implemented a prototype of the proposed techniques based on GNURadio and USRP, and performed real-world experiments to validate the proposed techniques. The experimental results showed that the proposed techniques reduce the reaction delay of the friendly jammer by 81.9% – 85.7%, and achieve accurate distinction between allies' and enemies' transmissions.

Wireless jamming attack is also used to launch the man-in-the-middle attack to wireless pairing. To address this problem, we presented a novel wireless in-band pairing design, in which the pairing device utilizes the 802.11 Clear to Send (CTS) reserved duration to detect the existence of possible attackers in the network. We further proposed to use the on/off sub-carriers in the frequency domain to convey the hash digest bits to protect the integrity of the benign pairing messages. Compared with the previous work, our design reduces most of the channel occupation time by eliminating the long synchronization packet and the on/off slots in the time domain. Our evaluation results showed that the proposed technique of using sub-carriers to convey hash bits works accurately in the real-world environment.

© Copyright 2015 by Wenbo Shen

All Rights Reserved

Physical Layer Signal Design and Control For Wireless Security

by  
Wenbo Shen

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

Computer Science

Raleigh, North Carolina

2015

APPROVED BY:

---

Dr. David Thuentle

---

Dr. Douglas Reeves

---

Dr. Peng Ning  
Co-chair of Advisory Committee

---

Dr. Huaiyu Dai  
Co-chair of Advisory Committee

## DEDICATION

To my parents and my wife.

## BIOGRAPHY

Wenbo Shen was born in a small village called Da Shen and was raised in Huaiyang, Henan Province, China. He completed his Bachelor degree in Software Engineering from Harbin Institute of Technology, in the year of 2010. He started his PhD study in Computer Science of North Carolina State University in August 2010. His research focuses on leveraging wireless signal properties to achieve wireless physical layer security.

## ACKNOWLEDGEMENTS

The completion of a PhD degree requires encouragement, support and help from many people.

First of all, I want to thank my co-advisors, Dr. Peng Ning and Dr. Huaiyu Dai. I have learned a lot from them, which is beyond the scope of academic. Again, I would like to thank them for their guidance, encouragement, patience and support. I also would like to thank my committee members Dr. David Thunte and Dr. Douglas Reeves for their insightful feedback to my research. Moreover, I also would like to thank my academic sister Dr. Yao Liu, thanks for her help, encouragement and guidance during my PhD study. I also want to thank my collaborator Xiaofan He, thanks for answering me so many questions about equations and signals.

Second, I would like to thank my friends for their generous help: Ruowen Wang, Yajin Zhou, Lei Wu, Pu Yang, Xianqing Yu, Syed Hussain, Wu Zhou, Dong Wang, Attila Yavuz, Ahmed Azab, Jason Gionta, Xi Ge, Entong Shen, Yinglei Zhang, Xiaopeng Duan, Wei Wang, Shen Ma, Xin Xu, Hongxia Chen, Shuangyu Xu and Kunsheng Fang.

Finally, I would like to thank National Science Foundation (NSF) for their funding support.

# TABLE OF CONTENTS

|                                                                                                                                              |               |
|----------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| <b>LIST OF TABLES</b> . . . . .                                                                                                              | <b>viii</b>   |
| <b>LIST OF FIGURES</b> . . . . .                                                                                                             | <b>ix</b>     |
| <b>Chapter 1 INTRODUCTION</b> . . . . .                                                                                                      | <b>1</b>      |
| 1.1 MCR Decoding . . . . .                                                                                                                   | 2             |
| 1.2 Ally Friendly Jamming . . . . .                                                                                                          | 3             |
| 1.3 Fast Physical Layer Verification of Friendly Jamming . . . . .                                                                           | 6             |
| 1.4 Efficient In-band Wireless Pairing through Specialized CTS and Multi-carrier<br>Communications . . . . .                                 | 7             |
| 1.5 Summary of Contributions . . . . .                                                                                                       | 9             |
| <br><b>Chapter 2 MCR Decoding: A MIMO Approach for Defending Against Wire-<br/>less Jamming Attacks</b> . . . . .                            | <br><b>11</b> |
| 2.1 Preliminaries . . . . .                                                                                                                  | 11            |
| 2.1.1 Wireless Communication Systems . . . . .                                                                                               | 11            |
| 2.1.2 MIMO Systems . . . . .                                                                                                                 | 12            |
| 2.2 Assumptions and Threat Model . . . . .                                                                                                   | 13            |
| 2.2.1 Assumptions . . . . .                                                                                                                  | 13            |
| 2.2.2 Threat Model . . . . .                                                                                                                 | 13            |
| 2.3 MCR Decoding . . . . .                                                                                                                   | 13            |
| 2.3.1 System Overview . . . . .                                                                                                              | 13            |
| 2.3.2 Transmission Detection . . . . .                                                                                                       | 15            |
| 2.3.3 Dealing With the Constant Jammer . . . . .                                                                                             | 17            |
| 2.3.4 Dealing with the Reactive Jammer . . . . .                                                                                             | 17            |
| 2.4 Analysis . . . . .                                                                                                                       | 19            |
| 2.4.1 MCR Processing Gain . . . . .                                                                                                          | 19            |
| 2.4.2 Bit Error Rate Analysis . . . . .                                                                                                      | 19            |
| 2.5 Experimental Evaluation . . . . .                                                                                                        | 21            |
| 2.5.1 Prototype Setup . . . . .                                                                                                              | 21            |
| 2.5.2 Evaluation . . . . .                                                                                                                   | 21            |
| 2.6 Related Work . . . . .                                                                                                                   | 23            |
| <br><b>Chapter 3 Ally Friendly Jamming: How to Jam Your Enemy and Maintain<br/>Your Own Wireless Connectivity at the Same Time</b> . . . . . | <br><b>25</b> |
| 3.1 Preliminaries . . . . .                                                                                                                  | 25            |
| 3.2 Assumptions and Threat Model . . . . .                                                                                                   | 27            |
| 3.3 Ally Friendly Jamming . . . . .                                                                                                          | 27            |
| 3.3.1 Generation of Ally Jamming Signals . . . . .                                                                                           | 28            |
| 3.3.2 Synchronizing with Ally Jamming Signals . . . . .                                                                                      | 30            |
| 3.3.3 The Introduction of Pilot Frequencies . . . . .                                                                                        | 32            |
| 3.3.4 Detecting and Recovering Transmissions . . . . .                                                                                       | 35            |
| 3.3.5 Dealing with Multiple Ally Jammers . . . . .                                                                                           | 38            |

|                                                                                                                        |                                                             |           |
|------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|-----------|
| 3.3.6                                                                                                                  | Dealing with Multiple Authorized Transmitters . . . . .     | 40        |
| 3.4                                                                                                                    | Analysis . . . . .                                          | 40        |
| 3.4.1                                                                                                                  | Maintaining Authorized Communication . . . . .              | 41        |
| 3.4.2                                                                                                                  | Disabling Unauthorized Communication . . . . .              | 42        |
| 3.4.3                                                                                                                  | JSR Trade-off . . . . .                                     | 43        |
| 3.4.4                                                                                                                  | Limitations . . . . .                                       | 44        |
| 3.5                                                                                                                    | Implementation and Evaluation . . . . .                     | 44        |
| 3.5.1                                                                                                                  | Experiment Setup . . . . .                                  | 44        |
| 3.5.2                                                                                                                  | Evaluation Methodology . . . . .                            | 45        |
| 3.5.3                                                                                                                  | Micro-Evaluation . . . . .                                  | 45        |
| 3.5.4                                                                                                                  | Macro-Evaluation . . . . .                                  | 48        |
| 3.6                                                                                                                    | Related Work . . . . .                                      | 50        |
| <br>                                                                                                                   |                                                             |           |
| <b>Chapter 4 No Time to Demodulate - Fast Physical Layer Verification of Friendly Jamming . . . . .</b>                |                                                             | <b>52</b> |
| 4.1                                                                                                                    | Preliminaries . . . . .                                     | 52        |
| 4.2                                                                                                                    | Assumptions and Threat Model . . . . .                      | 53        |
| 4.3                                                                                                                    | Fast Friendly Jamming . . . . .                             | 53        |
| 4.3.1                                                                                                                  | Overview . . . . .                                          | 53        |
| 4.3.2                                                                                                                  | Auth-Preamble Generation . . . . .                          | 54        |
| 4.3.3                                                                                                                  | Auth-Preamble Verification . . . . .                        | 55        |
| 4.4                                                                                                                    | Analysis . . . . .                                          | 58        |
| 4.4.1                                                                                                                  | Against Different Kinds of Attacks . . . . .                | 58        |
| 4.4.2                                                                                                                  | Against Anti-Jamming Unauthorized Devices . . . . .         | 61        |
| 4.4.3                                                                                                                  | Impact of Multiple Friendly Jammers . . . . .               | 61        |
| 4.4.4                                                                                                                  | Communication Overhead . . . . .                            | 61        |
| 4.5                                                                                                                    | Experimental Evaluation . . . . .                           | 62        |
| 4.5.1                                                                                                                  | Experiment Setup . . . . .                                  | 62        |
| 4.5.2                                                                                                                  | Auth-Preamble Verification Accuracy . . . . .               | 62        |
| 4.5.3                                                                                                                  | Execution Time . . . . .                                    | 63        |
| 4.6                                                                                                                    | Related Work . . . . .                                      | 64        |
| <br>                                                                                                                   |                                                             |           |
| <b>Chapter 5 Efficient In-band Wireless Pairing through Specialized CTS and Multi-carrier Communications . . . . .</b> |                                                             | <b>67</b> |
| 5.1                                                                                                                    | Preliminaries . . . . .                                     | 67        |
| 5.1.1                                                                                                                  | Wi-Fi Protected Setup and PBC . . . . .                     | 68        |
| 5.1.2                                                                                                                  | 802.11 and RTS/CTS . . . . .                                | 69        |
| 5.1.3                                                                                                                  | OFDM . . . . .                                              | 69        |
| 5.2                                                                                                                    | Assumptions and Threat Model . . . . .                      | 70        |
| 5.3                                                                                                                    | Protocol Design . . . . .                                   | 70        |
| 5.3.1                                                                                                                  | Tampering-Proof Pairing Message . . . . .                   | 71        |
| 5.3.2                                                                                                                  | Hiding-Proof Pairing Message . . . . .                      | 73        |
| 5.3.3                                                                                                                  | Distinguishing Attack Collisions from Normal Ones . . . . . | 75        |
| 5.3.4                                                                                                                  | Reducing Pairing Message Collisions . . . . .               | 76        |
| 5.3.5                                                                                                                  | Integrating with PBC . . . . .                              | 76        |

|                                                       |                                         |           |
|-------------------------------------------------------|-----------------------------------------|-----------|
| 5.3.6                                                 | Example Attack Scenarios . . . . .      | 78        |
| 5.4                                                   | Analysis . . . . .                      | 79        |
| 5.4.1                                                 | Channel Occupation Time . . . . .       | 79        |
| 5.4.2                                                 | Security Analysis . . . . .             | 80        |
| 5.5                                                   | Implementation and Evaluation . . . . . | 81        |
| 5.5.1                                                 | Prototype Setup . . . . .               | 81        |
| 5.5.2                                                 | Accuracy Evaluation . . . . .           | 81        |
| 5.6                                                   | Related work . . . . .                  | 84        |
| <b>Chapter 6 Conclusion and Future Work . . . . .</b> |                                         | <b>85</b> |
| 6.1                                                   | Conclusion . . . . .                    | 85        |
| 6.2                                                   | Future Work . . . . .                   | 86        |
| <b>References . . . . .</b>                           |                                         | <b>88</b> |

## LIST OF TABLES

|           |                         |    |
|-----------|-------------------------|----|
| Table 5.1 | Terminologies . . . . . | 71 |
|-----------|-------------------------|----|

## LIST OF FIGURES

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |    |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figure 1.1  | MITM attack scenarios. MSG is the benign pairing message while FMSG is the fake pairing message. Jam means the attacker is jamming the channel. Color (Or shadowed in black & white print) rectangles represent the transmitted messages, while un-colored ones represent the received messages. Due to the use of directional antenna, jamming signals and the transmitted FMSG below the time line can only be received by the access point while the ones above time line can only be received by the PBC device. . . . . | 8  |
| Figure 2.1  | A $2 \times 2$ MIMO system. $h_{ij}$ is the channel coefficient for transmitter antenna $i$ and receiver antenna $j$ , include channel attenuation and phase shift. . . . .                                                                                                                                                                                                                                                                                                                                                  | 12 |
| Figure 2.2  | A scenario with the jammer. The transmitter TX and the jammer are single-antenna devices, the receiver RX is a bi-antenna device. . . . .                                                                                                                                                                                                                                                                                                                                                                                    | 14 |
| Figure 2.3  | MCR values. (a) shows the case when only jammer is present; (b) shows MCR values when both the transmitter and jammer are transmitting. . . . .                                                                                                                                                                                                                                                                                                                                                                              | 16 |
| Figure 2.4  | Constant jamming scenario. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 17 |
| Figure 2.5  | Reactive jamming scenario. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 18 |
| Figure 2.6  | Bit error rate under different jamming power removal rates. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 20 |
| Figure 2.7  | False positive and true positive rates. The false positive rate is the detection rate when there are no transmissions. True positive rate is the detection rate when there is an ongoing transmission. . . . .                                                                                                                                                                                                                                                                                                               | 22 |
| Figure 2.8  | Jamming power removed by MCR decoding. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 23 |
| Figure 3.1  | Simplified structure for a wireless digital communication system. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 26 |
| Figure 3.2  | Illustration of ally friendly jamming. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 28 |
| Figure 3.3  | Generation of jamming signals. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 29 |
| Figure 3.4  | Synchronization with ally jamming signals. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 31 |
| Figure 3.5  | Received samples interpolation. Interpolation rate $N = 16$ . The selected interpolated samples are close to the received samples. . . . .                                                                                                                                                                                                                                                                                                                                                                                   | 33 |
| Figure 3.6  | Pilot frequency assignment. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 34 |
| Figure 3.7  | Received signal spectrum. Only show a portion of the whole spectrum. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 35 |
| Figure 3.8  | Transmission detection and recovery under ally friendly jamming. The authorized RX and the ally jammer are both in $i$ -th epoch. $s$ is the regenerated ally jamming signal, $y$ is the received ally jamming signal, $m$ is the received collided signal. $T$ is the re-synchronization interval. . . . .                                                                                                                                                                                                                  | 36 |
| Figure 3.9  | Estimated channel. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 37 |
| Figure 3.10 | Bit error rate analysis. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 41 |
| Figure 3.11 | Identifying ally friend jammers. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 46 |
| Figure 3.12 | Synchronizing with multiple ally jammers. The correlation length is 1000 samples. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                    | 47 |
| Figure 3.13 | Transmission detection rate. FP is the false positive rate, TP is the true positive rate. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                            | 48 |
| Figure 3.14 | Removal of ally jamming signals. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 48 |

|             |                                                                                                                                                                                                                                                                                                             |    |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figure 3.15 | Macro-evaluation. . . . .                                                                                                                                                                                                                                                                                   | 49 |
| Figure 3.16 | Jamming DSSS devices (kps: kilo symbols per second). . . . .                                                                                                                                                                                                                                                | 50 |
| Figure 4.1  | Jamming scenarios of fast friendly jamming. . . . .                                                                                                                                                                                                                                                         | 54 |
| Figure 4.2  | Transmitted auth-preamble signals, received auth-preamble signals and their amplitude differential values. . . . .                                                                                                                                                                                          | 56 |
| Figure 4.3  | Amplitude differential values and their bit representation. The chosen threshold is 1. . . . .                                                                                                                                                                                                              | 58 |
| Figure 4.4  | Replay attack scenario. AT is the ally transmission while UT is the unauthorized transmission. . . . .                                                                                                                                                                                                      | 60 |
| Figure 4.5  | True positive and false positive of amplitude differential based correlation. TP is true positive and FT is false positive. . . . .                                                                                                                                                                         | 63 |
| Figure 4.6  | True positive and false positive of efficient amplitude differential based correlation. . . . .                                                                                                                                                                                                             | 64 |
| Figure 4.7  | Time comparison. ADV is the amplitude differential values. . . . .                                                                                                                                                                                                                                          | 65 |
| Figure 5.1  | Push Button Configuration Pairing Process. . . . .                                                                                                                                                                                                                                                          | 67 |
| Figure 5.2  | Basic structure of an OFDM communication system. $b_i$ and $b'_i$ are the message bits, $s_i$ and $s'_i$ are the discrete base-band signals. . . . .                                                                                                                                                        | 68 |
| Figure 5.3  | Packet signal generation at the transmitter. . . . .                                                                                                                                                                                                                                                        | 71 |
| Figure 5.4  | Hash bits verification at the receiver. . . . .                                                                                                                                                                                                                                                             | 72 |
| Figure 5.5  | Pairing scenario with no attackers. The buttons on the enrollee and the registrar were pushed. . . . .                                                                                                                                                                                                      | 74 |
| Figure 5.6  | Pairing scenario with jamming attacks. The buttons on the enrollee and the registrar were pushed. The attacker uses directional antenna, so that the jamming signals below the time line can only be received by the registrar while the ones above time line can only be received by the enrollee. . . . . | 74 |
| Figure 5.7  | Pairing message format. The CTS, the payload signals and the hash signals are separated by a SIFS. . . . .                                                                                                                                                                                                  | 76 |
| Figure 5.8  | Attack scenarios. The buttons on the enrollee and the registrar were pushed in a) and b). . . . .                                                                                                                                                                                                           | 78 |
| Figure 5.9  | Hash signal generation. $s_i$ is the discrete base-band signal. . . . .                                                                                                                                                                                                                                     | 81 |
| Figure 5.10 | Hash bits and the sub-carrier energy. . . . .                                                                                                                                                                                                                                                               | 82 |
| Figure 5.11 | Energy CDF of on-off sub-carriers. . . . .                                                                                                                                                                                                                                                                  | 83 |
| Figure 5.12 | Hash bit error rates . . . . .                                                                                                                                                                                                                                                                              | 84 |

# Chapter 1

## INTRODUCTION

Wireless communication technology has been widely deployed and increasingly adopted due to the ease of installation and reduced operational cost. Various wireless applications, such as Wi-Fi, cellular networks and blue-tooth, are reshaping the way we live. As wireless devices become ubiquitous, more and more wireless attacks emerge. Due to the shared nature of wireless medium, wireless communication is very vulnerable to wireless jamming attacks. The attacker can simply use a wireless device to emit random wireless signals. As the wireless medium is shared, signals of the benign transmitter and the jammer will collide at the receiver, and the signal reception process is disrupted. Considering the fact that nowadays, the wireless hardware is becoming cheaper and easier to obtain, launching the wireless jamming attack has never been so easy before. On the other hand, recent advances of wireless technology also provide us new hopes for combating with the wireless jamming attack. For example, software-defined radio (SDR) provides the flexibility to manipulate wireless physical layer signals on PC, which allows us to study the physical layer signal properties thoroughly. Multiple-input and multiple-output (MIMO) technique uses multiple antennas to decode multiple concurrent transmissions, which inspired my first work - MCR Decoding, leveraging the multi-channel capability to defend against the wireless jamming attack.

While wireless jamming attack is a big threat to wireless communication, it also provides insights for researchers to design novel wireless techniques to protect wireless communication. Recently, friendly jamming (i.e., intentional signal interference from collaborating devices) has been proposed to protect information confidentiality as well as achieve wireless medium access control [75, 74]. One application of protecting information confidentiality is to protect the unencrypted wireless channels of legacy devices, so that eavesdropping will be defeated. Wireless medium access control applications can be blocking unauthorized wireless transmissions for RFID systems [67, 68] and implantable medical devices [91, 32]. Previous research works of using friendly jamming for access control cannot disable unauthorized wireless communications

in a region, which motivated my second work - Ally Friendly Jamming, which disables the adversary's wireless communication while still maintains our own wireless communication at the same time.

Moreover, when using reactive jamming for access control, the friendly jammer needs to first identify an on-going unauthorized wireless transmission and then launch jamming while the transmission is still on-the-air. Previous research studies rely on the bits information to distinguish the authorized and unauthorized transmissions, which requires the demodulation of the wireless signals. The demodulation process is time-consuming, and thus will introduce a non-trivial reaction delay, which will hurt the performance of friendly jamming. In order to solve this problem, we proposed Fast Friendly Jamming, which eliminates the need of demodulation and verifies the signals directly on the physical layer.

Wireless jamming is also used by the attacker to launch a variety of attacks, such as the man-in-the-middle attack to wireless pairing devices. With wireless devices becoming more and more popular, there is a demand of interconnecting them. To facilitate the pairing of Wi-Fi enabled devices, Wi-Fi Alliance introduced Push-Button Configuration (PBC). On the other hand, armed with cutting-edge techniques, the attacker can launch the man-in-the-middle attack by jamming to form a collision on the benign pairing transmission, and then impersonating one pairing party to pair with the other one. The previous defense scheme needs to occupy the channel continuously for a considerable period of time, which not only wastes the channel resource, but also hurts the transmissions from nearby wireless devices on the same channel. These problems motivated my fourth work - Efficient In-band Wireless Pairing through Specialized CTS and Multi-carrier Communications.

This dissertation contains these four works toward preserving wireless security. Details of these works will be shown in Chapters 2, 3, 4 and 5, respectively. In the following, I give motivations of these four works.

## 1.1 MCR Decoding

Due to the shared use of the wireless medium, wireless applications are vulnerable to jamming attacks. A jammer can simply emit random noise to disrupt wireless communications between the transmitter and the receiver. Therefore, the robustness against jamming attacks is crucial for wireless applications which require high communication reliability.

Spread spectrum techniques such as direct-sequence spread spectrum (DSSS), frequency-hopping spread spectrum (FHSS) and their enhanced variations, including the DSSS variations [48, 51, 64] and FHSS variations [47, 79], are commonly used techniques for anti-jamming wireless communications. However, these spread spectrum based schemes require large spectrum bandwidth, which is undesirable considering the scarcity of the wireless spectrum.

Recent advances of multiple-input and multiple-output (MIMO) technique [29] bring new hope for enhancing anti-jamming wireless communications. In particular, research groups have investigated the MIMO systems for interference cancellation. Gollakota et al. proposed the Technology Independent Multi-Output (TIMO) scheme [31], which exploits the channel ratio (the ratio of channel coefficients) of the interference source and the transmitter's to remove cross-technology interference for 802.11n wireless networks. Due to the requirement of the transmitter's channel state information, TIMO cannot deal with the fast reactive jamming attacks. Under the fast reactive jamming attack, the jamming signals start and stop at almost the same time with the desired transmission signals. The receiver has neither enough un-jammed preamble signals to estimate the transmitter's channel state information, nor pure jamming signals to compute the channel ratio for the jammer, and thus it cannot remove the jamming signals. Moreover, TIMO is only capable of removing interference from a single interference source, which is reasonable for dealing with unintended interference. However, for intended adversarial jammers, it is very likely that multiple jammers operate on the same frequency.

To address these problems, we extend the TIMO technique into the anti-jamming domain and propose an anti-jamming technique, *Multi-Channel Ratio (MCR) Decoding*, which exploits the multi-channel ratio and the repeated preamble signals to detect and recover the desired transmission signals under constant and reactive jamming attacks.

Unlike the spread spectrum schemes which suppress the jamming signals at the price of spectrum bandwidth, MCR decoding exploits the MCR to subtract the jamming signals directly. Moreover, MCR decoding does not require any shared keys to build its anti-jamming capability. Hence, it does not suffer from the vulnerabilities introduced by the shared keys.

Compared with TIMO, MCR decoding eliminates the need of the transmitter's channel state information, which makes it more suitable for anti-jamming applications. MCR decoding uses the multi-channel ratio to detect the transmissions under jamming attacks and can handle multiple constant jammers on the same frequency band as well as the fast reactive jammer, even though the reactive jamming signals start and stop at the same time with the desired transmission signals.

## 1.2 Ally Friendly Jamming

Wireless communication technology has been widely deployed and increasingly adopted due to the ease of installation and reduced operational cost. The applications that benefit from wireless communications range from traditional military operations to more recent civilian applications such as Wi-Fi and mobile phones. There have also been on-going efforts aimed at adopting wireless communications in emerging and mission-critical applications (e.g., health-care [32, 44] and critical infrastructure protection [18, 28]).

In mission-critical applications such as battlefield operations, anti-terrorism activities, and critical infrastructure protection, it is highly desirable and sometimes necessary to gain advantages over the adversary in terms of wireless communication capability. In particular, *it is highly desirable to disable the adversary's (unauthorized) wireless communication while still maintaining our own (authorized) wireless communication*. For example, wireless communications have been a common way to trigger Improvised Explosive Devices (IED) (a.k.a. roadside bombs), which were responsible for approximately 63% coalition deaths in the second Iraq war from 2001 to 2007 and over 66% of the coalition casualties in Afghanistan between 2001 and 2012 [4]. The capability of disabling enemy wireless communication and at the same time maintaining coalition's wireless connectivity would greatly reduce the casualties due to radio-controlled IED. It is conceivable that such a capability will also enhance the security of other non-military mission-critical applications such as critical infrastructure protection and health-care applications.

This work aims at providing such a capability. Specifically, we develop a novel mechanism, called *Ally Friendly Jamming*, to provide an intelligent jamming capability that can disable unauthorized (enemy) wireless communication but at the same time still allow authorized wireless devices to communicate, even if both the authorized and unauthorized devices operate at the same frequency.

The basic idea behind ally friendly jamming is to jam the wireless channel continuously but properly control the jamming signals using secret keys, so that the jamming signals are unpredictable interference to unauthorized devices, but are recoverable by authorized devices equipped with the secret keys. As a result, when authorized devices need to communicate, they can employ proper signal processing techniques to remove the jamming signals and recover the messages transmitted by other authorized devices. In other words, authorized devices can regenerate jamming signals using the secret keys and subtract them from the received, mixed signals to get jamming-free transmissions.

Though conceptually simple, ally friendly jamming turns out to be non-trivial to achieve. We have to resolve three technical challenges to ensure effective jamming and at the same time enable authorized devices to actually receive messages under ally friendly jamming, even though such devices know the secret keys.

First, to achieve ally friendly jamming, the ally jamming signals need to be irresolvable interference to unauthorized devices. Simply transmitting modulated pseudo random numbers as jamming messages can be easily defeated due to the strong patterns introduced by the digital communication process (e.g., modulation) [36]. Thus, the jamming signals injected by ally jammers must resemble real random noises. In the proposed ally friendly jamming scheme, we introduce the concept of *epoch* and use the shared keys with epoch indices as the input of a pseudo random number generator to directly control physical layer symbols, so that these signals are random noises to unauthorized devices and easy for authorized devices to synchronize with.

Second, an authorized receiver has to synchronize with the ally jammers, so that it can estimate the ally jamming signals, remove them from received signals, and recover potential transmissions from authorized transmitters. Though synchronization is a well-studied problem in digital communication, synchronization in ally friendly jamming faces a new challenge. As the channel and hardware effects (e.g., frequency offset) on the received ally jamming signals are unknown, the authorized receiver cannot synchronize with ally jammers even though it can generate the same transmitted ally jamming signals. The frequency offset can be compensated for by using the phase-locked loop which depends on the strong phase patterns existing in the transmitted signals. As ally jamming signals mimic random noises, no strong patterns can be relied on, so existing synchronization approaches (e.g., [37, 65, 30, 57]) cannot be applied directly in ally friendly jamming. In this work, we propose to use the pilot frequency aided correlation to synchronize authorized receivers with multiple ally jammers.

Third, when multiple ally jammers exist in the network, an authorized receiver needs to first identify these ally jammers properly and then regenerate the transmitted ally jamming signals in order to recover the authorized transmission. A particular challenge lies in how to identify these ally jammers rapidly while their ally jamming signals are pseudo-random signals and the channel and hardware effects on the received ally jamming signals are unknown. To solve this problem, we propose to use the pilot frequency and the fast Fourier transform (FFT) to identify ally jammers and further compensate for the hardware difference effects on the received signals.

A similar technique called IMD (Implantable Medical Device) shield [32] was proposed recently which exploited jamming to provide access control to an IMD. The IMD shield is a small radio device that employs two antennas for jamming and receiving, respectively. The receive antenna is physically connected to a transmit (jam)-and-receive chain, so that when sending a jamming signal, the jam chain can inject an “antidote” signal to the receive antenna to cancel the jamming signal. Due to the physical connection between the jamming and the receiving antennas, IMD shield does not have to deal with the synchronization challenge addressed in this work. Moreover, the multiple-jammer case was not considered in IMD shield. This means if multiple IMD shields operate at the same time in the same area, their jamming signals will interfere with each other, and all accesses will be denied. Therefore, by providing solutions to the above problems, our work further advances the current state of the art in security enhancement through friendly jamming.

We have implemented a prototype for ally friendly jamming using the Universal Software Radio Peripheral (USRP) platform [10] and GNURadio [2]. Our experimental results show that under ally friendly jamming, authorized devices have close-to-0 packet loss rate, and at the same time unauthorized devices suffer from 100% packet loss rate.

### 1.3 Fast Physical Layer Verification of Friendly Jamming

Besides continuous jamming, which is used by ally friendly jamming, reactive jamming is another way to achieve wireless medium access control. By using reactive jamming, the friendly jammer needs to block unauthorized wireless transmissions and avoid jamming the authorized ones mistakenly. In other words, the friendly jammer needs to identify the on-going wireless transmission first, and keeps silent if it is authorized or launches jamming attacks otherwise.

To achieve effective jamming, the friendly jammer needs to identify and jam an unauthorized wireless transmission while the transmission is still on the air. Thus the reaction time is crucial to the jamming performance. Previous friendly jamming studies (e.g., [67, 68, 91, 32, 85, 86]) proposed to distinguish wireless transmissions by using bit-level information, such as matching certain patterns in the message bits. However, in order to obtain the message bits, the friendly jammer needs to perform signal demodulation, which normally involves cascading steps, such as frequency offset compensation, symbol synchronization, and constellation decoding. These steps impose a non-trivial time delay for the friendly jammer, and thus the friendly jammer may fail to identify the unauthorized transmissions in a timely manner. Therefore, to reduce the reaction time, we propose *fast friendly jamming*, which eliminates the need of demodulation and verifies the signals directly on the physical layer.

The basic idea of fast friendly jamming is that the authorized transmitter generates a special preamble (that we name as *auth-preamble*, short for authentication preamble) using a shared secret key and prepends the auth-preamble before the packet transmission (i.e., before the normal preamble). On the other side, the friendly jammer uses the same key to synchronize and verify the auth-preamble of an on-going transmission. If the verification succeeds, the friendly jammer will keep silent; otherwise, the current on-going transmission will be treated as an unauthorized one and the friendly jammer will launch jamming.

Though conceptually simple, two technical challenges need to be solved before achieving fast friendly jamming. First, to eliminate the demodulation steps and allow the direct verification of the auth-preamble on the physical layer, the auth-preamble signals cannot be modulated bits<sup>1</sup>. Moreover, the auth-preamble signals must introduce enough randomness and should be ever changing to prevent the adversary from predicting, mimicking and replaying them. To address these problems, we propose to use the shared secret key and the time info to generate signal symbols in the auth-preamble directly.

Second, the auth-preamble signals introduce randomness to defend against the predict, mimic and replay attacks. However, the randomness also brings difficulties for the friendly jammer to verify the auth-preamble. Simple correlation won't work as the frequency offset will

---

<sup>1</sup>In this work, modulation refers to the base-band modulation, in which bits are mapped to points on a constellation diagram. The modulated base-band signals still need to be up-converted to radio frequency band before being sent out from an antenna.

distort the correlation results [75]. Traditional synchronization approaches [65, 30] cannot be applied for the friendly jammer to synchronize with the auth-preamble signals, as the auth-preamble signals are changing continuously and the channel and hardware effects (i.e., channel attenuation, phase shift, and frequency offset) on the received signals are unknown. To address this problem, we propose a novel technique called *amplitude differential based correlation*, which can tolerate the unknown channel and hardware effects on the received signals, thereby allowing the friendly jammer to verify the received signals directly on the physical layer.

## 1.4 Efficient In-band Wireless Pairing through Specialized CTS and Multi-carrier Communications

Wireless technologies have been widely deployed in recent decades. A common application of wireless technologies is the 802.11 Wi-Fi network with a variety of Wi-Fi enabled wireless devices. While users are increasingly concerned about the wireless security, there is a proliferation of Wi-Fi enabled devices with very simple user interfaces that don't support strong security configuration (e.g., no user interface for entering a key) due to limitations on the cost, design or functionality. To facilitate the use of these Wi-Fi enabled devices, Push-Button Configuration (PBC) was introduced by the Wi-Fi Alliance, as a mandatory part of Wi-Fi Protected Setup (WPS) [11]. PBC allows devices with very simple user interfaces and no out-of-band channels (e.g., key input interface, infra-red channel, camera or microphone) to generate common shared keys to protect their wireless communications [12].

PBC uses the Diffie-Hellman key exchange [26] to protect wireless communications from eavesdropping attacks. However, it is vulnerable to man-in-the-middle (MITM) attacks. For example, when a PBC device wants to pair with the access point, the attacker can launch two types of MITM attacks, as shown in Fig. 1.1.

In Scenario 1, the directional antenna and full-duplex radio techniques [17] are used by the attacker to achieve selective jamming and jamming-and-receiving. The attacker first jams the benign pairing message to form a collision at the access point, then it impersonates the access point to pair with the PBC device. After that, it uses the same trick to pair with the access point. As a result, the MITM attack is successful. While in Scenario 2, the attacker tampers the content of a benign pairing message by transmitting FMSG at the same time with the sender, but with a much stronger power, to produce a capture effect at the receiver, so that the transmission from the sender becomes noise and the receiver will accept the pairing message from the attacker.

From the above two scenarios, we can see that the root cause of the MITM attack to PBC is the lack of authentication of pairing parties. Considering that nowadays, PBC is increasingly

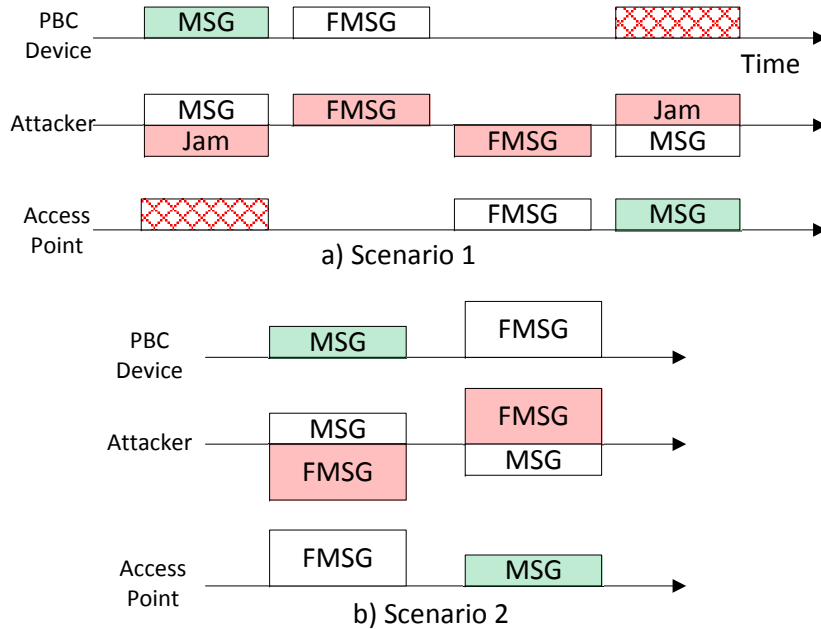


Figure 1.1: MITM attack scenarios. MSG is the benign pairing message while FMSG is the fake pairing message. Jam means the attacker is jamming the channel. Color (Or shadowed in black & white print) rectangles represent the transmitted messages, while un-colored ones represent the received messages. Due to the use of directional antenna, jamming signals and the transmitted FMSG below the time line can only be received by the access point while the ones above time line can only be received by the PBC device.

adopted by wireless medical devices which transmit patients' private health information [5, 13] and home security devices such as home surveillance cameras [3, 1], the MITM attack to PBC can be a severe threat to users' privacy, while PBC may give users a false sense of security [35, 42].

To defend against the MITM attack in PBC, TEP [35] proposed to use an exceptionally long synchronization packet and the on-off slots in the time domain to protect the existence and the integrity of a pairing message. For example, to prevent the attacker from hiding the pairing message by jamming to form a collision, TEP utilizes a synchronization packet which is longer than any collisions in Wi-Fi networks, so that collisions introduced by attack jamming can be distinguished from normal collisions. To detect any tampering attempts to the benign pairing message, TEP uses on-off slots in the time domain to convey pairing message hash digest from the sender to the receiver, in which bit "1" is represented by the *on slot* (i.e., slot with energy) and bit "0" is represented by the *off slot* (i.e., slot with no energy). As the attacker cannot cancel out the energy in the on slot, by equalizing the numbers of on and off slots, any

tampering attempts to modify the benign pairing message can be detected.

However, the security comes with a price. TEP has two negative impacts on nearby wireless devices. First, the long channel continuous occupation of a TEP message may interrupt wireless connections of nearby devices. For example, each TEP pairing message needs to occupy the busy 2.4 GHz channel for more than 24,760  $\mu s$  continuously, which may cause the TCP re-transmission or even TCP dis-connection. Second, TEP wastes channel resources and harms the Wi-Fi throughput. In PBC, the joining device (i.e., enrollee) keeps sending pairing messages to scan all channels for about 120 seconds. The repeatedly transmitted TEP pairing messages and the responses will occupy the busy 2.4 GHz channel for a considerable amount of time during that 120 seconds, even though there might be no attacker at all. Considering that more and more medical devices are using 2.4 GHz channels to communicate with the Android phones [58] or other monitoring devices and the fact that 2.4 GHz band is overcrowded, the long channel occupation time of the TEP message is undesirable.

To reduce the channel occupation time, we propose to use a specially crafted Clear to Send (CTS) request and the cooperation between the pairing devices to eliminate the need of the long synchronization packet in TEP. Our key observation is that the benign devices will respect the CTS requests and will keep silent during the CTS reserved duration. On the contrary, in order to launch the MITM attack, the attacker has to jam the pairing packet even though it received the CTS packet. Thus, if one pairing device sends out the CTS successfully and detects collisions in the CTS reserved duration, it deduces that attackers may exist in the network and hence will abort the pairing process to avoid pairing with the attacker falsely. Moreover, we propose to use OFDM sub-carriers in the frequency domain as the on-off slots. As the 802.11 a/g OFDM implementation supports 52 sub-carriers, the transmitter can map the hash bits of the pairing message to different sub-carriers, in which bit “1” is mapped to an *on sub-carrier* (i.e., sub-carrier with energy), while the bit “0” is mapped to an *off sub-carrier* (i.e., sub-carrier with no energy). The receiver de-maps the sub-carriers to hash bits by using the energy detection to distinguish the on/off sub-carriers. With these two techniques, each pairing message in our design occupies the channel for about 487  $\mu s$ , which reduces more than 98% of channel occupation time comparing with TEP.

## 1.5 Summary of Contributions

The contributions of this dissertation are summarized below:

- MCR Decoding: First, we identify the limitations of applying the TIMO technique into the anti-jamming domain and propose MCR decoding which can detect and recover the desired transmission signals under jamming attacks. Second, we have implemented a pro-

prototype for MCR decoding based on the GNURadio [2] and Universal Software Radio Peripheral (USRP) [10], and performed extensive experimental evaluations. Our experimental results show that MCR decoding can detect the desired transmission accurately under the jamming attack and remove more than 99.86% of the jamming signal power.

- **Ally Friendly Jamming:** We explore a new concept called ally friendly jamming that can disable unauthorized wireless communication and at the same time allow authorized devices to maintain wireless connectivity. We develop new techniques to generate ally jamming signals, to identify and synchronize with multiple ally jammers. We have also implemented a prototype for ally friendly jamming and performed analysis and extensive experimental evaluation to validate the techniques.
- **Fast Friendly Jamming:** First, we propose fast friendly jamming as well as the related techniques, which enable the friendly jammer to verify the received signals directly on the physical layer. Second, we have implemented a prototype of the proposed techniques on GNURadio [2] and USRP [10], and performed real-world experiments to validate the proposed techniques. The experimental results show that fast friendly jamming reduces the reaction delay of the friendly jammer by 81.9%–85.7%, as compared to the traditional demodulation approach. Meanwhile, it enables the accurate distinction between allies' and enemies' transmissions with 100% true positive and 0% false negative rates.
- **Efficient In-band Wireless Pairing through Specialized CTS and Multi-carrier Communications:** The contributions of this work are two-fold. First, we propose a secure wireless in-band pairing design by using the CTS and the cooperation between pairing devices to prevent the attacker from hiding the pairing message, and using on-off sub-carriers in the frequency domain to protect the pairing message integrity. Our design reduces most of the channel occupation time of TEP and provides the same security guarantee. Second, we have implemented and evaluated the proposed techniques based on GNURadio [2] and USRP [10]. The real world experiments show that the proposed technique of using sub-carriers to convey hash digest bits works accurately.

## Chapter 2

# MCR Decoding: A MIMO Approach for Defending Against Wireless Jamming Attacks

### 2.1 Preliminaries

#### 2.1.1 Wireless Communication Systems

Wireless communication systems generally use radio frequency (RF) signals to convey information. Upon receiving bits from upper layers, the transmitter first maps them to *discrete base-band signals* (a.k.a. *physical layer symbols*), then converts these discrete signals to analog signals, and finally up-converts them to RF signals [75].

The RF signals go through the wireless channel before being received by the receiver. The wireless channel introduces attenuation, phase shift, and noise during transmission. The hardware of the transmitter and the receiver introduces the frequency offset  $\Delta f$  [33]. Thus after the signal  $x(i)$  is transmitted through the channel, it is transformed into the received signal  $y(i)$ , and

$$y(i) = he^{j2\pi\Delta ft_i}x(i) + n(i)^1,$$

where  $h$  is a complex number, containing *channel attenuation* and *phase shift*,  $e^{j2\pi\Delta ft_i}$  is a complex number in its polar form (i.e., a complex number  $a + bj$  can be represented by its polar form  $Me^{j\theta}$ , where  $M = \sqrt{a^2 + b^2}$  and  $\theta = \tan^{-1}(b/a)$  [53]),  $n(i)$  is the *noise* and  $t_i$  is the sampling time for sample  $y(i)$ ;  $y(i)$  and  $x(i)$  are discrete base-band signals, which can also be represented by complex numbers.

---

<sup>1</sup>This equation is for single-tap channels.

Even though channel effects and the frequency offset are unknown, the receiver can use the phase-locked loop to compensate  $\Delta f$ , and use differential encoding schemes to tolerate the phase shift. Therefore, it can recover the transmitted signal  $x(i)$  by using existing synchronization approaches [65, 30, 45]. Then the receiver can de-map the physical layer symbol  $x(i)$  to bits and recover the transmitted message.

### 2.1.2 MIMO Systems

In a MIMO system, if both transmitting and receiving antennas are separated properly, the channel between each transmitting and receiving antennas pair will be different from each other [81]. Consider a  $2 \times 2$  MIMO system shown in Fig. 2.1, the transmitter sends signals of

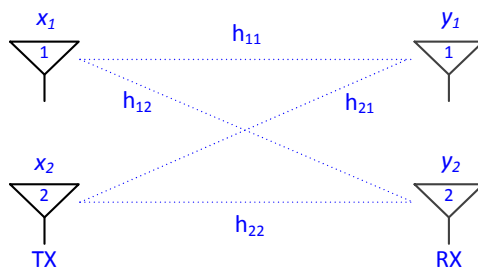


Figure 2.1: A  $2 \times 2$  MIMO system.  $h_{ij}$  is the channel coefficient for transmitter antenna  $i$  and receiver antenna  $j$ , include channel attenuation and phase shift.

two packets,  $x_1$  on Antenna 1,  $x_2$  on Antenna 2 concurrently, the receiver will receive

$$\begin{cases} y_1(i) = h_{11} \cdot x_1(i) + h_{21} \cdot x_2(i) + n_1(i) \\ y_2(i) = h_{12} \cdot x_1(i) + h_{22} \cdot x_2(i) + n_2(i) \end{cases},$$

where  $y_m(i)$  is the signal received by the  $m$ th antenna of the receiver,  $n_1(i)$  and  $n_2(i)$  are the white noise. As the signal to noise ratio (SNR) is high enough, if the receiver knows the channel coefficients  $h_{ij}$ , it can solve the above equations (two equations and two unknowns  $x_1(i)$  and  $x_2(i)$ ) to decode the concurrently transmitted packets.

To let the receiver compute the channel coefficients, the MIMO transmitter starts each frame by transmitting a known preamble from each of its antennas, one after the other [31]. By combining the knowledge of both the received and the transmitted preamble signals, the receiver can compute the channel coefficients, which can be used to decode the packet signals of this frame [57, 22]. However, if the jammer jams the preamble, the received preamble signals

will be totally disrupted by the jamming signals, which will lead to wrong estimations of the channel coefficients. As a result, the received signals cannot be decoded.

## 2.2 Assumptions and Threat Model

### 2.2.1 Assumptions

We assume that the wireless channels are single-tap. As the receiver needs and both the received transmission signals and the received jamming signals at the receiver have a sufficient signal to noise ratio (SNR). We assume all devices, including the transmitter, the receiver and the jammer, are immobile, hence, channels between them do not change significantly in a short period. Finally, we assume the receiver has at least two antennas while the jammer and the transmitter have single antenna. We will generalize MCR decoding for the multi-antenna jammer in our future work.

### 2.2.2 Threat Model

The objective of the jammer is to defeat the proposed scheme to disable the legal wireless transmissions. MCR only depends on the channel coefficients, rather than the jamming signals, which makes it only sensitive to whether the jamming is on or not. The jammer can use different strategies to jam channel. According to the jamming strategies, we classify the jammers into three categories: the constant jammer, the random on-off jammer and the reactive jammer.

The constant jammer emits random jamming signals all the time. The random on-off jammer jams the channel or keeps silent for random intervals. The reactive jammer listens to the channel and transmits jamming signals when an ongoing communication is detected.

The anti-jamming scheme and the analysis of MCR decoding for the constant jammer and the random on-off jammer are roughly the same. For brevity, we only consider the constant jammer and reactive jammer in the following sections.

## 2.3 MCR Decoding

In this section, we first give an overview of MCR decoding, then discuss the techniques against the constant jammer and the reactive jammer.

### 2.3.1 System Overview

The basic idea of MCR decoding is to exploit the channel ratio of the jammer (i.e., the ratio of two channel coefficients) to remove the jamming signals, then use certain techniques such as differential encoding, phase-locked loop to recover the desired transmission signals. Let us

use the scenario shown in Fig. 2.2 to illustrate the process. The receiver does not know the

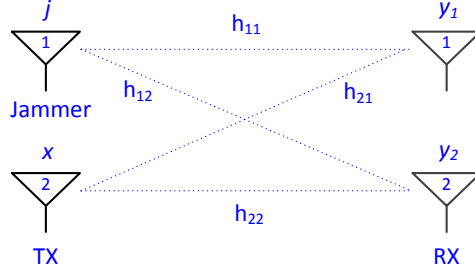


Figure 2.2: A scenario with the jammer. The transmitter TX and the jammer are single-antenna devices, the receiver RX is a bi-antenna device.

jammer’s channels  $h_{11}$  and  $h_{12}$ . Under the jamming attack, the receiver cannot even estimate the transmitter’s channels  $h_{21}$  and  $h_{22}$ . Therefore, the traditional MIMO approaches cannot be applied to recover the transmitter’s signals under the jamming attack even though the receiver has two antennas.

To defend against the jamming attacks, MCR decoding uses the multi-channel ratio (MCR) to remove the jamming signals. Here we assume the jammer is a constant jammer and defer the discussion on the reactive jammer case to later sections. Assume the transmitter is silent, the jammer is jamming and for the jamming signal  $j(i)$  emitted by the jammer, the received signal by receiver’s two antennas are  $y_1(i)$  and  $y_2(i)$  respectively, ignoring the white noise, we have

$$\begin{cases} y_1(i) = h_{11}e^{j2\pi\Delta f_j t_i} \cdot j(i) \\ y_2(i) = h_{12}e^{j2\pi\Delta f_j t_i} \cdot j(i) \end{cases},$$

where  $\Delta f_j$  is the frequency offset between the receiver and the jammer<sup>2</sup> and  $t_i$  is the sampling time. We use  $\varphi$  to represent the MCR of the two channels between the receiver and the jammer, and thus

$$\varphi(i) = \frac{h_{11}}{h_{12}} = \frac{y_1(i)}{y_2(i)}. \quad (2.1)$$

It is worth noting that  $\varphi$  does not rely on the jamming signals. In other words, if the jammer and the receiver do not move,  $\varphi$  should remain the same over a short period time (e.g., several ms).

Then when the transmission is being jammed, assume the transmitter TX is transmitting

---

<sup>2</sup>The signal processing blocks of these two receiving antennas share the same clock source, so the frequency offsets are the same.

signal  $x(k)$ , while the jammer is emitting signal  $j(k)$ , the received signals from the two antennas are  $y_1(k)$ ,  $y_2(k)$  respectively, we have

$$\begin{cases} y_1(k) = h_{11}e^{j2\pi\Delta f_j t_k} \cdot j(k) + h_{21}e^{j2\pi\Delta f t_k} \cdot x(k) \\ y_2(k) = h_{12}e^{j2\pi\Delta f_j t_k} \cdot j(k) + h_{22}e^{j2\pi\Delta f t_k} \cdot x(k) \end{cases}. \quad (2.2)$$

As  $\varphi$  remains the same for a short time, we can replace  $h_{11}$  in Equation (2.2) with  $\varphi(i)$  and  $h_{12}$  so that

$$\begin{cases} y_1(k) = h_{12}e^{j2\pi\Delta f_j t_k} j(k) \cdot \varphi(i) + h_{21}e^{j2\pi\Delta f t_k} x(k) \\ y_2(k) = h_{12}e^{j2\pi\Delta f_j t_k} j(k) + h_{22}e^{j2\pi\Delta f t_k} x(k) \end{cases}.$$

As the value of  $\varphi(i)$  is known, the receiver can treat  $h_{12}e^{j2\pi\Delta f_j t_k} j(k)$  as one unknown and subtract it from  $y_1(k)$  or  $y_2(k)$ , then it follows that

$$[\varphi(i) \cdot h_{22} - h_{21}]e^{j2\pi\Delta f t_k} x(k) = \varphi(i)y_2(k) - y_1(k),$$

where  $y_1(k)$ ,  $y_2(k)$  and  $\varphi(i)$  are known. Even though  $h_{21}$  and  $h_{22}$  are unknown, the receiver can treat  $[\varphi(i) \cdot h_{22} - h_{21}]e^{j2\pi\Delta f}$  as the new channel coefficient so that it can use phase-locked loop to compensate the effect of  $e^{j2\pi\Delta f}$ . The differential encoding at both the transmitter and the receiver can tolerate the phase shift introduced by  $\varphi(i) \cdot h_{22} - h_{21}$ . Therefore, by regarding  $[\varphi(i) \cdot h_{22} - h_{21}]e^{j2\pi\Delta f}$  as the new, unknown channel efficient, the receiver can tolerate its impacts and recover  $x(k)$  under the jamming attack.

### 2.3.2 Transmission Detection

To reduce the work load, the receiver only needs to perform MCR decoding when it detects the being jammed transmissions. Therefore, it needs to be able to detect the ongoing transmissions under the jamming attack. It turns out that this problem can be solved by monitoring MCR values, and we term this technique as *MCR Detection*. To simplify the analysis, we assume the jammer here is the constant jammer. The reactive jammer case can be treated similarly.

The intuition of MCR detection is that when only jammer is transmitting, the estimated MCR values are stable over a short period. In contrast, when the ongoing transmission collides with the jamming signals, the estimated MCR values will change significantly.

Considering the scenario in Fig. 2.2, when only the jammer is transmitting, we can compute MCR value  $\varphi(i) = \frac{h_{11}}{h_{12}} = \frac{y_1(i)}{y_2(i)}$  as the Equation (2.1) shows.  $\varphi$  only depends on the channels between the jammer and the receiver, which will be stable in a short time.

When the transmitter TX starts to transmit, the received signals  $y_1$  and  $y_2$  contain both jamming and the transmitter's signals, as shown by Equation (2.2). If the receiver uses the

same way (i.e., Equation (2.1)) to compute the MCR value, then the MCR value of the  $k$ -th sample becomes

$$\varphi(k) = \frac{h_{11}e^{j2\pi\Delta f_j t_k} \cdot j(k) + h_{21}e^{j2\pi\Delta f t_k} \cdot x(k)}{h_{12}e^{j2\pi\Delta f_j t_k} \cdot j(k) + h_{22}e^{j2\pi\Delta f t_k} \cdot x(k)}.$$

The stability of MCR is disrupted by the transmitter's signal components (i.e.,  $h_{21}e^{j2\pi\Delta f t_k}x(k)$ ,  $h_{22}e^{j2\pi\Delta f t_k}x(k)$ ). Therefore, the receiver can measure the standard deviation of  $\varphi$ 's amplitudes and use it as an indicator. If the standard deviation is greater than a certain threshold, a jammed transmission is detected by the receiver.

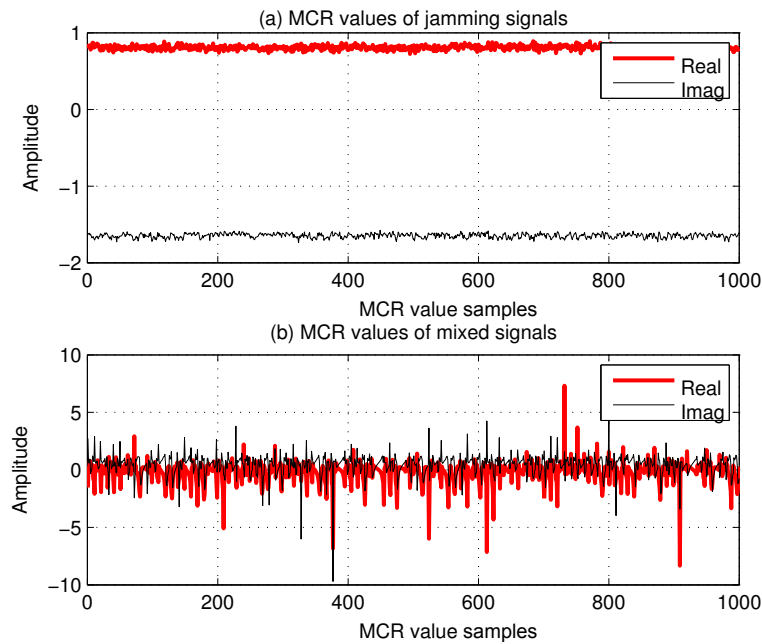


Figure 2.3: MCR values. (a) shows the case when only jammer is present; (b) shows MCR values when both the transmitter and jammer are transmitting.

Fig. 2.3 shows  $\varphi$  values obtained in our experiments. It is easy to see that when only jammer is present, MCR values are stable. On the contrary, when both the jammer and the transmitter are working, MCR values change significantly from time to time.

### 2.3.3 Dealing With the Constant Jammer

The constant jammer jams the channel all the time to disable any wireless communications. To defeat the constant jamming attack, the receiver can first use MCR detection to detect the transmission boundary. Then, as shown in Fig. 2.4, it can use the received signals which contain

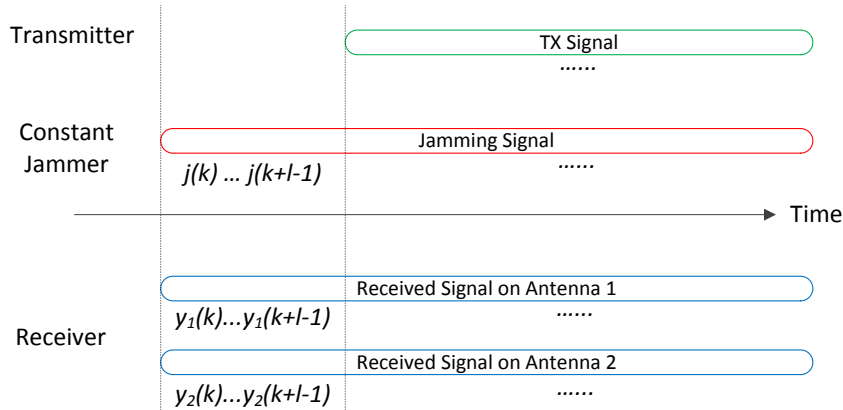


Figure 2.4: Constant jamming scenario.

no transmission signals (i.e.,  $y_1(k), \dots, y_1(k+l-1)$  and  $y_2(k), \dots, y_2(k+l-1)$ ) to compute jammer's MCR value  $\varphi$ , and then apply  $\varphi$  to remove the jamming signals in the following received signal samples. Therefore, the transmission signals can be recovered even under the jamming attack.

Note that the above discussion is for single constant jammer case. When multiple constant jammers (i.e.,  $n$  jammers) exist in the network and start to jam the channel at different time, the receiver which equips  $n+1$  antennas can use the above approach to remove the jamming signals from each jammer iteratively. Relevant discussion is omitted to avoid redundancy.

### 2.3.4 Dealing with the Reactive Jammer

For a fast reactive jammer, the reaction delay is very short, and the reactive jamming signals will always co-exist with the desired transmission signals. We term this kind of jamming attack as the fast reactive jamming attack and it cannot be defended by applying the TIMO technique. Therefore, to defeat the fast reactive jamming attack, we propose to use repeated preambles in the transmissions. As the same preamble signals will be transmitted twice, the receiver can exploit the repeated preamble signals to remove the transmission signals so that jammer's MCR can be computed.

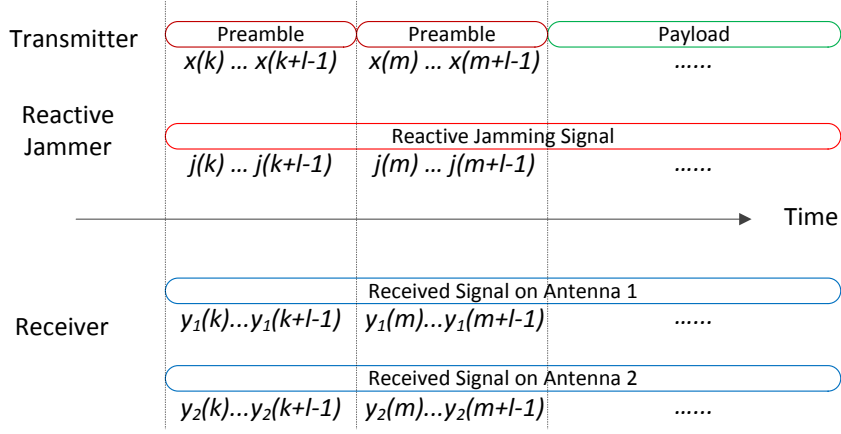


Figure 2.5: Reactive jamming scenario.

As shown in Fig. 2.5, the reactive jamming starts and stops at the exactly the same time with the desired transmission. For the transmitted signals of the first preamble, due to the reactive jamming, the receiver obtains

$$\begin{cases} y_1(i) = h_{11}e^{j2\pi\Delta f_j t_i} \cdot j(i) + h_{21}e^{j2\pi\Delta f t_i} \cdot x(i) \\ y_2(i) = h_{12}e^{j2\pi\Delta f_j t_i} \cdot j(i) + h_{22}e^{j2\pi\Delta f t_i} \cdot x(i) \end{cases}, \quad (2.3)$$

where  $i \in [k, \dots, k+l-1]$ . Then, for the transmitted preamble signals of the second preamble, the receiver gets

$$\begin{cases} y_1(n) = h_{11}e^{j2\pi\Delta f_j t_n} j(n) + h_{21}e^{j2\pi\Delta f t_n} x(n) \\ y_2(n) = h_{12}e^{j2\pi\Delta f_j t_n} j(n) + h_{22}e^{j2\pi\Delta f t_n} x(n) \end{cases}, \quad (2.4)$$

where  $n \in [m, \dots, m+l-1]$ . The receiver needs to remove the preamble signals  $x(i)$  or  $x(n)$  so that it can compute the jammer's multi-channel ratio. However, due to the frequency offset, subtraction cannot be done directly [75] even though  $x(i) = x(n)$ . The receiver needs to find a way to compute the frequency offset by using the jammed preamble signals.

In MCR decoding, the receiver uses the Frequency-Domain Correlation and Matching technique (FDCM) [93] to get an estimation of the frequency offset. The key observation of FDCM is that the exponential change on a sequence of signals in the time domain becomes linear in the frequency domain [93]. In other words, if the receiver does a Discrete Fourier Transform (DFT) on the  $l$  received preamble signals, due to the frequency offset  $\Delta f$ , all the DFT values will be shifted by  $\Delta f$ . Thus, by correlating the original DFT values and the shifted values,  $\Delta f$  can be estimated by finding the correlation peak. As the DFT also has the linearity property,

this approach can get the  $\Delta f$  even if the received signals are the mixture of the preamble signals and the jamming signals. After getting  $\Delta f$ , the receiver can multiple Equation (2.4) by  $\alpha = e^{j2\pi\Delta f(t_i - t_n)}$  and subtract Equation (2.4) from Equation (2.3) to remove  $x(i)$  and  $x(n)$ , then we have

$$\begin{cases} y_1(i) - \alpha y_1(n) = h_{11}[e^{j2\pi\Delta f_j t_i} j(i) - e^{j2\pi\Delta f_j t_n} \alpha j(n)] \\ y_2(i) - \alpha y_2(n) = h_{12}[e^{j2\pi\Delta f_j t_i} j(i) - e^{j2\pi\Delta f_j t_n} \alpha j(n)] \end{cases}.$$

Consequently, the jammer's MCR can computed as

$$\phi = \frac{h_{11}}{h_{12}} = \frac{y_1(i) - \alpha y_1(n)}{y_2(i) - \alpha y_2(n)}.$$

The computed MCR value  $\phi$  can be used for removing the jamming signal components in the following received jammed signals.

## 2.4 Analysis

In this section, we first analyze the processing gain of MCR decoding, then discuss bit error rate of the receiver when different percentage of jamming power is removed.

### 2.4.1 MCR Processing Gain

By removing the jamming signal power, MCR decoding provides the processing gain for the multi-antenna devices. Assume  $x$  is the percentage of the jamming signal power which is removed by MCR decoding, and  $G_m$  is the MCR processing gain, then we have

$$G_m = \frac{1}{1 - x}.$$

In our experiments, MCR decoding can remove more than 99.86% of the jamming signal power, then we can derive that  $G_m = 28.5$  dB. Note that when working with other anti-jamming schemes (e.g., DSSS), the MCR processing gain can add up with other processing gains. In other words, MCR decoding provides 28.5 dB extra anti-jamming capacity in addition to other anti-jamming schemes, which can be used to defeat the high power jamming attacks.

### 2.4.2 Bit Error Rate Analysis

Here we assume that the receiver only uses MCR decoding for anti-jamming. We use the bit error rate of the receiver to measure the effectiveness of MCR decoding against jamming attacks.

Let us first clarify some notations. We denote the power of received jamming signal, received transmission signal and noise are  $J$ ,  $R$  and  $N$  respectively. Thus, the jamming to signal power

ratio is  $JSR = \frac{J}{R}$ , the signal to noise ratio is  $SNR = \frac{R}{N}$ .

According to [30], the bit error rate (BER) of a wireless device is dependent on its SNR and the modulation method, as  $x$  percent of the jamming signal power can be removed by MCR decoding, then we can derive the BER for binary phase shift keying (BPSK) as

$$P_e = Q\left(\sqrt{\frac{2}{\frac{1}{SNR} + JSR(1-x)}}\right),$$

where  $Q(\cdot)$  is the Q-function [8].

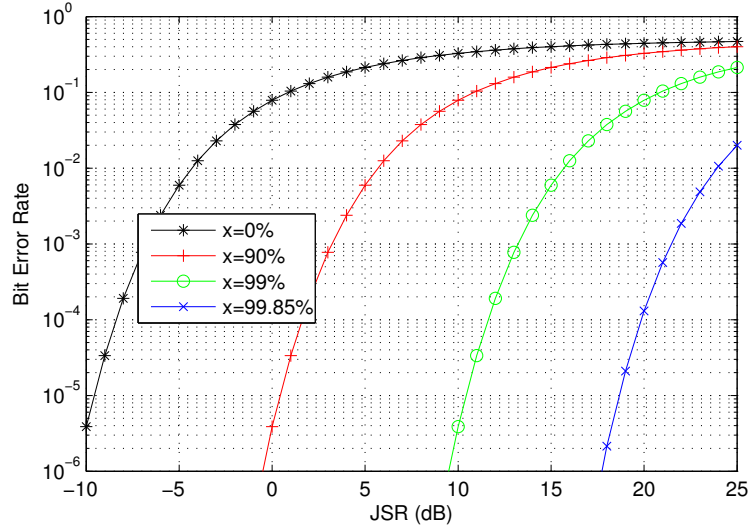


Figure 2.6: Bit error rate under different jamming power removal rates.

Fig. 2.6 gives the BER values w.r.t.  $x$  and JSR (with sufficient  $SNR$ ,  $\frac{1}{SNR}$  is close-to-0). It is generally agreed that a packet can be received correctly when its BER is less than  $10^{-3}$  [33], then from Fig. 2.6, we can see that when  $x = 99.85\%$ , the packets can be received correctly even though the jamming signal is 21 dB stronger than the transmission signal, as the vast majority of the jamming power is removed. Note that we use BPSK for modulation in the analysis, the results for other modulation methods can be derived similarly.

## 2.5 Experimental Evaluation

We have built a prototype for MCR decoding based on GNURadio and the USRP platform, and performed the real world experiments to validate our proposed techniques. In our experiments, we first validate the accuracy of MCR detection, then evaluate the removal of the jamming signal.

### 2.5.1 Prototype Setup

**Hardware Configuration:** The prototype system consists of a jammer, a transmitter, and a MIMO receiver; the jammer and transmitter are implemented using the USRP-N210 board connected to a host laptop via 1 Gbps Ethernet cable. The MIMO receiver is built by connecting two USRP-N210 boards with a MIMO cable. The USRP-N210 board uses a XCVR2450 daughter board operating on the 2.4GHz band as its RF front end. The MIMO receiver is about two meters away from the jammer and the transmitter.

**Software Configuration:** The jamming symbol rate for the jammer is  $5 \times 10^5$  samples per second (sps). The transmitter and the receiver use the differential binary phase shift keying for modulation and use both GNURadio and MATLAB for signal processing.

### 2.5.2 Evaluation

#### Transmission Detection

In the experiments, we first start the jammer and the transmitter, adjust their gains to achieve 0 dB, 5 dB, 10 dB, 15 dB and 20 dB JSR. The jammer keeps on jamming the channel while the transmitter is transmitting. Then we start the MIMO receiver, which samples the wireless channel at a rate of  $10^6$  sps and saves the samples in a file for subsequent processing.

We use the standard deviation of 500 MCR values' amplitudes to detect ongoing transmissions. By choosing different threshold values, we get the true positive and false positive rates of MCR detection as shown in Fig. 2.7. Here true positive means there is a transmission and the receiver detects it; while false positive means there is no transmission, but the receiver detects one mistakenly. It is easy to see that there is a range of threshold values which allow the transmissions to be detected almost 100% with close-to-0 false positive rate, even when the jamming signal strength is 20 dB stronger than the desired transmission signal strength. Therefore, the proposed MCR detection can detect the desired transmission accurately under jamming attacks.

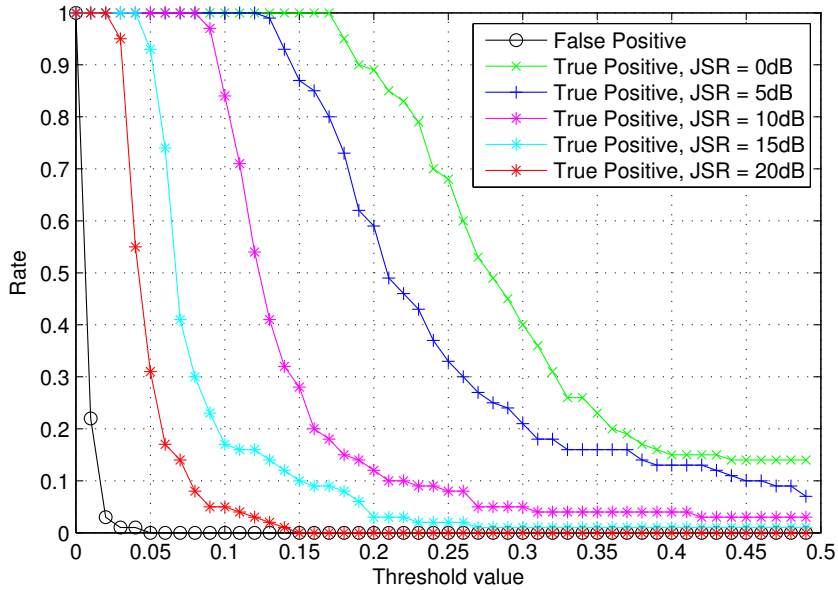


Figure 2.7: False positive and true positive rates. The false positive rate is the detection rate when there are no transmissions. True positive rate is the detection rate when there is an ongoing transmission.

### The Removal of Jamming Signal

In this part of experiments, we evaluate the jamming signal removing performance. We only start the jammer and the receiver. The jammer jams the channel all the time. The receiver first records the received jamming signals into a file, uses the first 1000 signal samples to compute the MCR value of the jammer, and then use the computed MCR to eliminate the jamming signals in the following signal samples.

In our experiments, the percentage of jamming power that can be removed by MCR decoding depends on how many samples we need to apply the elimination. The reason is that the channels between the jammer and receiver are changing slightly over time. If we apply the same MCR value to do elimination on too many samples, the difference between the MCR value we use and the real MCR value will become larger, thus less jamming power can be removed. Fig. 2.8 shows that when the sample number is from 1,000 to 15,000, more than 99.86% jamming signal power is removed. In other words, the vast majority of the jamming signal power can be effectively removed by MCR decoding.

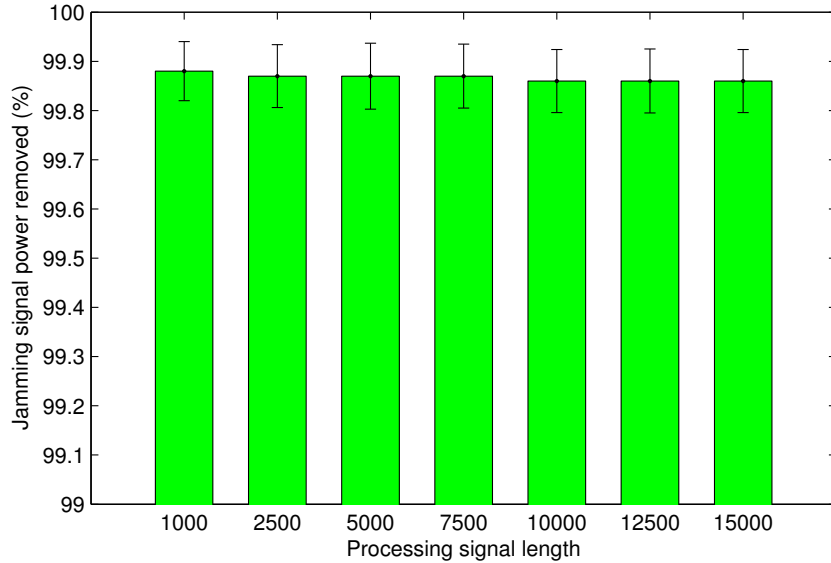


Figure 2.8: Jamming power removed by MCR decoding.

## 2.6 Related Work

Our work is related to anti-jamming and MIMO based interference cancellation techniques.

In [14], Aryafar et al. designed and implemented a multi-user beam-forming system and an experimental MIMO framework for wireless LANs. The Interference Alignment and Cancellation (IAC) technique was proposed in [34] to enable collaborative Access Points (APs) in MIMO LANs to decode more packets by controlling transmitted signals with proper vectors. In SAM [81], Tan et al. proposed a chain-decoding scheme which uses interference nullifying and cancellation to decode concurrent frames. It requires all stations to coordinate their transmissions so that the chain-decoding can be achieved. 802.11n<sup>+</sup> [46] proposed to use “antidote” signals to nullify the transmitted signals from other nodes in order to enable multiple access to wireless channels. All these three schemes require nodes to coordinate their transmissions so that the receiver can obtain the sender’s channel coefficients, which makes them unsuitable for anti-jamming purpose. TIMO [31] reported techniques which exploits the channel ratio of the interference source to remove cross-technology interference for 802.11n. However, it requires that the receiver knows the channel information of the transmitter, which is not feasible under fast reactive jamming attack.

For the anti-jamming techniques, traditional DSSS and FHSS anti-jamming schemes are vulnerable due to the shared keys. In recently years, researchers have developed the corre-

sponding enhanced schemes, including the DSSS variations [48, 51, 64, 63] and FHSS variations [47, 78, 79, 80]. All these schemes are orthogonal to and thus can be combined with our method proposed in this work.

## Chapter 3

# Ally Friendly Jamming: How to Jam Your Enemy and Maintain Your Own Wireless Connectivity at the Same Time

### 3.1 Preliminaries

Wireless digital communication systems generally employ radio frequency (RF) signals to transmit information. Transmitters need to convert digital messages represented in bits to RF signals, while receivers convert received RF signals back to digital messages. Figure 3.1 shows a simplified structure for a wireless digital communication system with one transmitter and one receiver. On the transmitter side, upon receiving bits from upper layers, the transmitter first modulates them to discrete baseband signals (a.k.a. *physical layer symbols*, or simply *symbols*), then converts them to analog signals using a digital to analog converter (DAC), and finally up-converts them to RF signals. The RF signals go through the wireless channel and reach the receiver. Upon receiving the RF signals, the receiver performs the inverse processing. It down-converts and samples the received signals to discrete baseband signals, and then demodulates them to bits.

Physical layer symbols are represented by complex numbers. For example, when BPSK is used for modulation, the transmitter modulates bit “1” to  $x = 1+0j$  and bit “0” to  $x' = -1+0j$  ( $j$  is the imaginary unit, satisfying  $j^2 = -1$ ). A symbol  $x_i = a + bj$  is often represented in its polar form  $x_i = Me^{j\theta}$ , where  $M = |x_i| = \sqrt{a^2 + b^2}$  and  $\theta = \tan^{-1}(b/a)$  [53].

The wireless channel introduces attenuation, phase shift, and additional noise during trans-

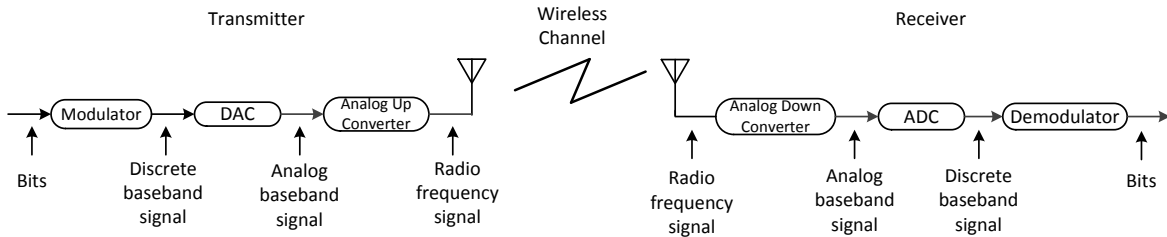


Figure 3.1: Simplified structure for a wireless digital communication system.

mission. After the signal  $x_i$  is transmitted through the channel, it is transformed into the received signal

$$y_i = h e^{j\gamma} x_i + n_i,$$

where  $h$  is the *channel attenuation*,  $\gamma$  is the *phase shift*, and  $n_i$  is the *noise*.

In practice, the signal reception at the receiver is also affected by two additional factors: *frequency offset* and *sampling offset*. Frequency offset  $\Delta f$  generally exists between the transmitter and the receiver, since there is no practical way to guarantee that two radios operate at exactly the same frequency.  $\Delta f$  causes variations on the phases of received signals [33]. Thus, if we take  $\Delta f$  into consideration, the received signal becomes

$$y_i = h e^{j\gamma} e^{j2\pi\Delta f t_i} x_i + n_i, \quad (3.1)$$

where  $t_i$  is the time at which the receiver gets the sample  $y_i$ .

Moreover, the receiver uses sampling and quantizing to recover the original baseband signals. Due to the lack of perfect synchronization in wireless communications, the receiver usually cannot sample perfectly to get the exact physical layer symbols sent by the transmitter. When the sampling offset is considered, the received signal becomes

$$y_i = h e^{j\gamma} e^{j2\pi\Delta f t_i} x_{i+\mu} + n_i, \quad (3.2)$$

where  $\mu$  is the sampling offset due to mis-sampling.

In summary, the wireless channel and the hardware differences introduce various distortion to the signal transmission. To correctly recover the transmitted messages, the receiver need to either estimate these parameters to certain accuracy or tolerate their influences.

## 3.2 Assumptions and Threat Model

**Assumptions:** We assume that there are multiple ally jammers and multiple authorized wireless devices, all of which share a secret key set that is unknown to unauthorized devices. We assume a high signal-to-noise ratio (SNR) for both transmission signals and ally jamming signals at the receiver. We also assume that the clocks at ally jammers and authorized devices are loosely synchronized, and the frequency offsets between ally jammers and authorized devices are within a given range. We assume that ally jammers can block the operational frequencies of all devices, including both authorized and unauthorized devices. In other words, unauthorized devices cannot find a wireless communication channel that is not being jammed by the ally jammers. We also assume that the adversary cannot defeat ally friendly jamming by physically removing ally jammers. Finally, we assume that each device (authorized or unauthorized) is equipped with a single omni-directional antenna and there is no adversarial jammer. How to accomplish ally friendly jamming with MIMO (multiple-input and multiple-output) devices and how to maintain wireless communication under both ally and adversarial jamming will be addressed in our future work.

**Threat Model:** We consider unauthorized devices as potential adversaries. The objective of unauthorized devices is to defeat the proposed scheme so that they can communicate under ally friendly jamming. They may analyze the ally friendly jamming signals and attempt to use the result of analysis to remove the jamming signals with signal processing techniques (e.g., [24, 25]). They may also employ anti-jamming communication techniques such as Direct Sequence Spread Spectrum (DSSS), Frequency Hopping Spread Spectrum (FHSS), and their variations (e.g., [63, 79, 47]).

## 3.3 Ally Friendly Jamming

In ally friendly jamming, upon detecting a transmission, the authorized device can employ proper signal processing techniques to remove the jamming signals from the received, mixed signals. In contrast, the unauthorized device does not have the secret keys, and cannot remove the interference introduced by ally jamming signals.

Figure 3.2 further illustrates ally friendly jamming, where one ally jammer is presented for simplicity. Assuming the ally jammer, the authorized and unauthorized devices are all in the same area. As mentioned earlier, the ally jammer and authorized devices, including  $A_1$ ,  $A_2$ , and  $AJ$  in Figure 3.2, share a secret key  $k$ . The ally jammer  $AJ$  uses a Pseudo-Random Number Generator (PRNG) with  $k$  as the seed to continuously emit jamming signals  $X_J$ .

When the unauthorized device  $E_1$  transmits signals  $X_{E_1}$  to another unauthorized device  $E_3$ , the signals received by  $E_3$  will be the mixture of both  $X_{E_1}$  and some portion of  $X_J$ . With

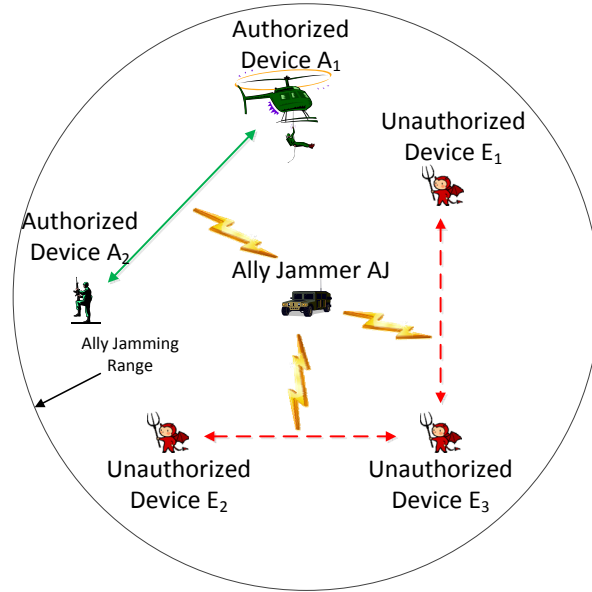


Figure 3.2: Illustration of ally friendly jamming.

enough jamming power, the jamming signals from  $AJ$  can effectively distort the signals  $X_{E_1}$  at  $E_3$ . As a result, the wireless communication between unauthorized devices  $E_1$  and  $E_3$  is disabled.

When  $A_1$  transmits signals  $X_{A_1}$  to  $A_2$ , the jamming signals  $X_J$  will also distort the received signals at  $A_2$ . However, since  $A_2$  shares the same secret key  $k$  with  $AJ$ , it can regenerate the same jamming signals  $X_J$  using  $k$ . If it can find out which portion of  $X_J$  is mixed with  $X_{A_1}$ , it can subtract this portion of  $X_J$  to get a clean copy of  $X_{A_1}$ . To remove  $X_J$  from the mixed signals, authorized devices need to synchronize with the ally jamming signals, estimate their values in the mixed signals, and remove them from the received, mixed signals to recover meaningful transmissions.

In the following sections, we will present how the ally jammer generates ally jamming signals and how the authorized device synchronizes with ally jammers and recovers the transmissions.

### 3.3.1 Generation of Ally Jamming Signals

Every ally jammer uses a shared, unique secret key to generate its ally jamming signals. Ally jammers and authorized devices share a set of secret keys. Either group key agreement (e.g., [41, 49, 88]) or group key distribution protocols (e.g., [61, 50, 23]) can be used to generate the secret key set. Assuming there are  $n$  ally jammers in the network, identified as  $AJ_1, AJ_2, \dots, AJ_n$  and  $n$  keys  $k_1, k_2, \dots, k_n$  in the key set, the key  $k_g$  will be assigned to the ally jammer  $AJ_g$ .

To ensure effective jamming against unauthorized devices, the jamming signals injected by ally jammers should resemble random noises. To achieve this goal, we use a PRNG to directly control the physical layer symbols so that these signals appear to be random noises to unauthorized devices. Since a physical layer symbol is represented as a complex number, we can use a PRNG to generate random floating point numbers with certain precision as the real and the imaginary parts of each symbol.

Moreover, the injected jamming signals should allow the authorized devices, which have access to the secret keys, to synchronize with ally jammers, even they join the network in the middle of a jamming session and the jamming has been going on for a long period of time.

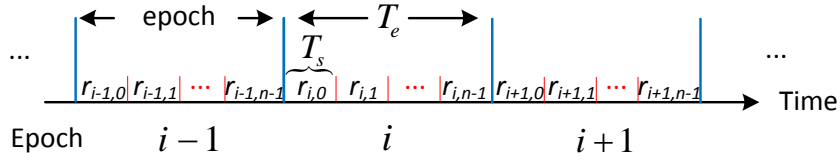


Figure 3.3: Generation of jamming signals.

To accomplish these goals, we make the following design, illustrated in Figure 3.3. We divide the time into equal-sized epochs, each of which consists of  $n$  physical layer symbols. Assuming that the duration of each physical layer symbol is  $T_s$ . Then the duration of each epoch is  $T_e = n \cdot T_s$ . For simplicity, we consider the time period  $[i \cdot T_e, (i + 1) \cdot T_e)$  as the  $i$ -th epoch, where  $i$  is the epoch index. For convenience, we also index and label the physical layer symbols within each epoch. For example, in Figure 3.3, the symbols in the  $i$ -th epoch are indexed from 0 to  $n - 1$  and labeled as  $r_{i,0}$  through  $r_{i,n-1}$ . With this design, for any given time  $t$ , we can easily compute the corresponding epoch index as  $i = \lfloor \frac{t}{T_e} \rfloor$ , and the symbol index within the epoch as  $m = \lfloor \frac{t - i \cdot T_e}{T_s} \rfloor$ . The corresponding physical layer symbol is thus  $r_{i,m}$ .

To allow easy synchronization with the jamming signals on authorized devices, we propose to use both the secret key and the epoch index to control the PRNG for jamming signal generation. Specifically, to generate the jamming symbols in epoch  $i$ , the ally jammer, say  $AJ_g$ , first uses the key  $k_g$  and the epoch index  $i$  as the seed to the PRNG to get a sequence of pseudo random floating numbers, i.e.,  $\langle a_0, a_1, \dots, a_{2n-1} \rangle = PRNG(k_g, i)$ , and then forms each jamming symbol  $r_{i,m}$  as  $r_{i,m} = a_{2m} + a_{2m+1} \cdot j$ , where  $m = 0, 1, \dots, n - 1$ . As a result, the jamming signals are pseudo-random samples, which are independent of the noise and shifted versions of themselves. Therefore, when an authorized device comes to the network, it can refine its synchronization with the ally jammer, and eventually remove the jamming signals.

Note that the quality of the jamming signals is affected by two parameters: the duration of each jamming symbol  $T_s$ , and the precision of the pseudo random numbers used for the real and the imaginary parts of jamming symbols. To maximize the uncertainty of the jamming signals, the smallest value for  $T_s$  and the maximum precision allowed for the jamming symbols can be used. Both parameters are eventually limited by the hardware used for emitting jamming signals. Finally, to ensure the randomness, the jamming symbols should be transmitted without modulation and encoding.

### 3.3.2 Synchronizing with Ally Jamming Signals

#### Synchronizing by Correlation

An authorized device has to synchronize with ally jammers, so that it can estimate and remove the ally jamming signals to maintain its communication. The goal of synchronization is to align the received ally jamming symbols with the locally generated ally jamming signals, even though these received signals have been distorted by the unknown wireless channel parameters (i.e., when the parameters  $\gamma$ ,  $\Delta f$ , and  $\mu$  in Equation (3.2) are unknown).

Let us use Figure 3.4 to explain the synchronization process in ally friendly jamming. In this and the following two sections, we will focus on one ally jammer for simplicity, and defer the discussion of multiple ally jammers to the Section 3.3.5. Assuming when an authorized device joins the network, the ally jammer, say  $AJ_g$ , is in the  $i$ -th epoch on its local clock and the ally jamming signals being transmitted are  $r_{i,k}, \dots, r_{i,l}$ . The corresponding jamming signals received by the authorized device are  $y_{i,k}, \dots, y_{i,l}$ . Assuming the frequency offset between  $AJ_g$  and the authorized device is  $\Delta f_g$ , based on Equation (3.1), we have

$$y_{i,m} = h e^{j\gamma} e^{j2\pi\Delta f_g t_{i,m}} r_{i,m} + n_{i,m}, m \in [k, l].$$

At the same time, the authorized device is in the  $(i + \delta)$ -th epoch on its own clock ( $\delta = -2$  in Figure 3.4). Assuming the authorized device knows that the ally jammer is  $AJ_g$  (we will address how to distinguish ally jammers in Section 3.3.3), it can use the secret key  $k_g$  and its epoch indices to regenerate the ally jamming symbols locally. It is assumed that the ally jammer and authorized devices are loosely synchronized, with maximum clock difference of  $\Delta T$ . Thus, the current local epochs of this authorized device and the ally jammer will not be more than  $w = \lceil \frac{\Delta T}{T_c} \rceil$  epochs away from each other, and the authorized device only needs to consider possible symbol alignments within this time window. In our example, since the authorized device is in the  $(i - 2)$ -th epoch, it should regenerate the following sequence of jamming symbols from the ally jammer:  $r_{d,0}, r_{d,1}, \dots, r_{d,n-1}$ , where  $d \in [i - 2 - w, i - 2 + w]$ .

To obtain the synchronization with the ally jammer, the authorized device can use correla-

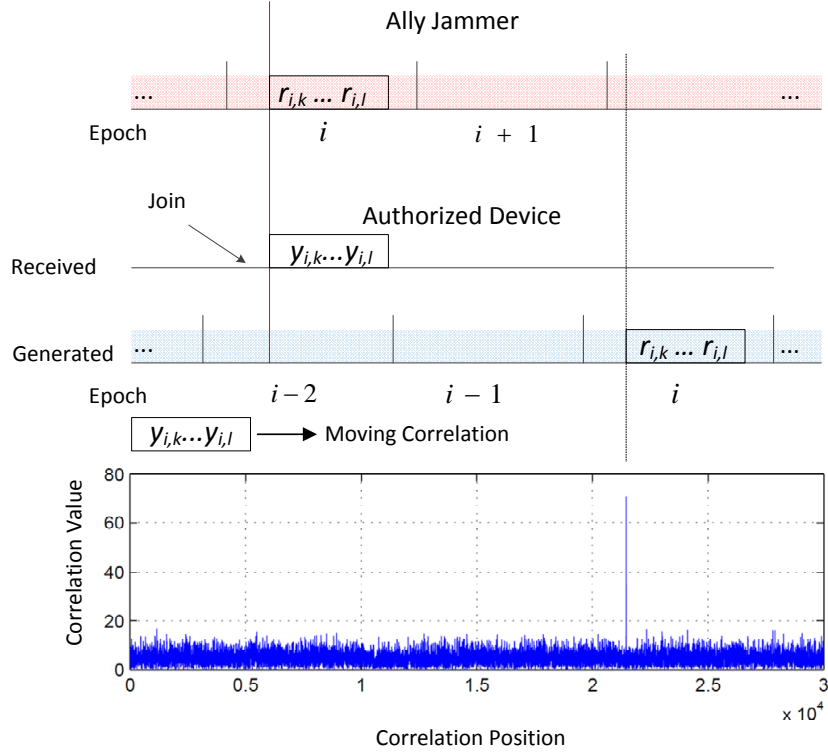


Figure 3.4: Synchronization with ally jamming signals.

tion to find the location of the received samples  $y_{i,k}, \dots, y_{i,l}$  in the locally generated symbols. Correlation is a popular technique for detecting known signal patterns on the receiver side. Assuming the correlation length is  $L$ . The authorized device can firstly align  $y_{i,k}, \dots, y_{i,k+L-1}$  with the first  $L$  signals in  $r_{d,0}, r_{d,1}, \dots, r_{d,n-1}$ , compute the correlation, shift the alignment by one sample and re-compute the correlation, until a spike at the correlator output is identified. The jamming signals are pseudo-random samples, which are independent of the noise and shifted versions of themselves. Therefore, the correlation is near zero except when the correct alignment is found.

However, the above statement is only partially correct as the frequency offset can disrupt the correlation. For example, assuming the correlation output is  $\Gamma$ :

$$\begin{aligned}
 \Gamma &= \sum_{n=0}^{L-1} y_{i,k+n} \cdot r_{i',k'+n}^* \\
 &= \sum_{n=0}^{L-1} [h e^{j\gamma} e^{j2\pi\Delta f_g t_{i,k+n}} r_{i,k+n} + n_{i,k+n}] \cdot r_{i',k'+n}^*
 \end{aligned}$$

where  $r_{i',k'+n}$  is a signal in the locally generated jamming signal sequence  $r_{d,0}, r_{d,1}, \dots, r_{d,n-1}$  and  $r_{i',k'+n}^*$  is its complex conjugation. As  $r_{i',k'+n}^*$  is independent of noise,  $n_{i,k+n}$  will be canceled out. If the correct alignment is found, say  $i' = i$  and  $k' = k$ , then we have

$$\Gamma \approx h e^{j\gamma} \sum_{n=0}^{L-1} |r_{i,k+n}|^2 e^{j2\pi\Delta f_g t_{i,k+n}}.$$

The frequency offset part  $e^{j2\pi\Delta f_g t_{i,k+n}}$  introduces dynamic phases to the individual components in the above sum, which may lead to signal cancellation. Therefore, the authorized device must compensate for frequency offset before the correlation can be used for synchronization. After compensating for the frequency offset (we will discuss frequency offset compensation in 3.3.3), the correlation output becomes:

$$\begin{aligned} \Gamma &\approx h e^{j\gamma} \sum_{n=0}^{L-1} |r_{i,k+n}|^2 e^{j2\pi\Delta f_g t_{i,k+n}} \cdot e^{-j2\pi\Delta f_g t_{i,k+n}} \\ &\approx h e^{j\gamma} \sum_{n=0}^{L-1} |r_{i,k+n}|^2. \end{aligned}$$

The correlation spikes when the received signals are aligned correctly with the generated signals, as shown in Figure 3.4. Therefore, by detecting the correlation spike, the authorized device is able to synchronize with the ally jammer.

Recall that there is also a sampling offset between the received ally jamming signals and the self-generated signals. For example, assuming for any transmitted jamming signal  $r_{i,m}$ , the received signal by authorized device with sampling offset  $\mu$  is  $r_{i,m+\mu}$ . After generating  $r_{i,m}$  with the shared key, the authorized device interpolates it at a rate of  $N$ . As a result,  $r_{i,m}$  will be expanded to  $r_{i,m+p/N}, p = 0, \dots, N - 1$ . When  $N$  is large enough (in our experiments,  $N = 16$  gives a good enough resolution), there will be a value  $p_0$  such that  $p_0/N \approx \mu$ , as shown in Figure 3.5. The authorized device can use  $p_0/N$  to approximate the sampling offset  $\mu$ .

To decide the value of  $p_0$ , the authorized device uses a selection of the interpolated samples rather than the samples before interpolation, to correlate with the received signals. The authorized device can try all values of  $p = 0, 1, \dots, N - 1$ , the one achieving the maximum correlation spike value is regarded as  $p_0$ , which can be used to approximate the sampling offset for the following samples.

### 3.3.3 The Introduction of Pilot Frequencies

In order to compensate for the frequency offset as well as identify ally jammers rapidly, we introduce the concept *pilot frequency* into ally friendly jamming. A pilot frequency is a 1 Hz

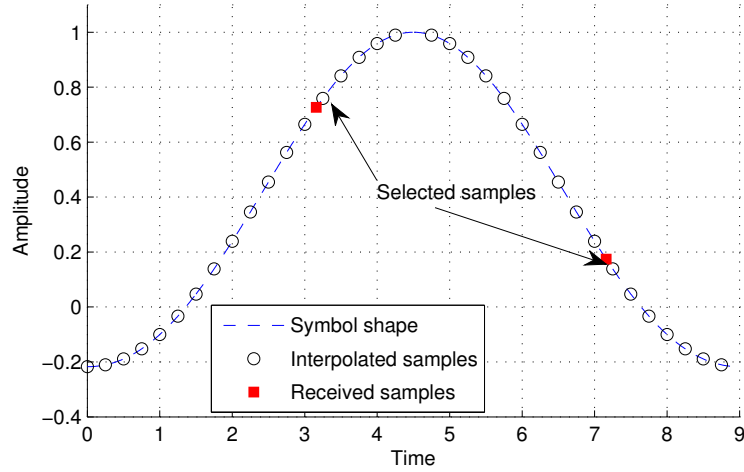


Figure 3.5: Received samples interpolation. Interpolation rate  $N = 16$ . The selected interpolated samples are close to the received samples.

wide frequency uniquely associated with each ally jammer, injected along with the pseudo-random jamming signals into the channel. On the receiver side, the authorized device can use this pilot frequency to identify the associated ally jammer and compute the frequency offset between them.

Before applying pilot frequency, we need to assign a proper pilot frequency to each ally jammer. Assuming the maximum frequency offset between ally jammers and authorized devices is  $f_{max}$ , the frequency offset  $\Delta f \in [0, f_{max})$ . We assign  $(2g - 1)f_{max}$  as the ally jammer  $AJ_g$ 's pilot frequency and designate  $[(2g - 2)f_{max}, 2gf_{max})$  as the associated shift range, as shown in Figure 3.6.

For each ally jammer, along with the generated pseudo-random signals, it also generates the signals of its pilot frequency. Assume an epoch has  $n$  pseudo-random signals, the ally jammer will generate  $n$  pilot frequency signals, and apply them to all epochs. For example, for the ally jammer  $AJ_g$  with the pilot frequency  $(2g - 1)f_{max}$ , the pilot frequency signal it will generate for the  $m$ -th pseudo-random signals in all epochs is

$$pf_m = e^{j2\pi(2g-1)f_{max}mT_s}.$$

$pf_m$  will be added up onto the  $m$ -th generated pseudo-random signals in all the epochs. Hence the  $m$ -th jamming signals in epoch  $i$ , say  $s_{i,m}$ , is given by

$$s_{i,m} = r_{i,m} + pf_m.$$

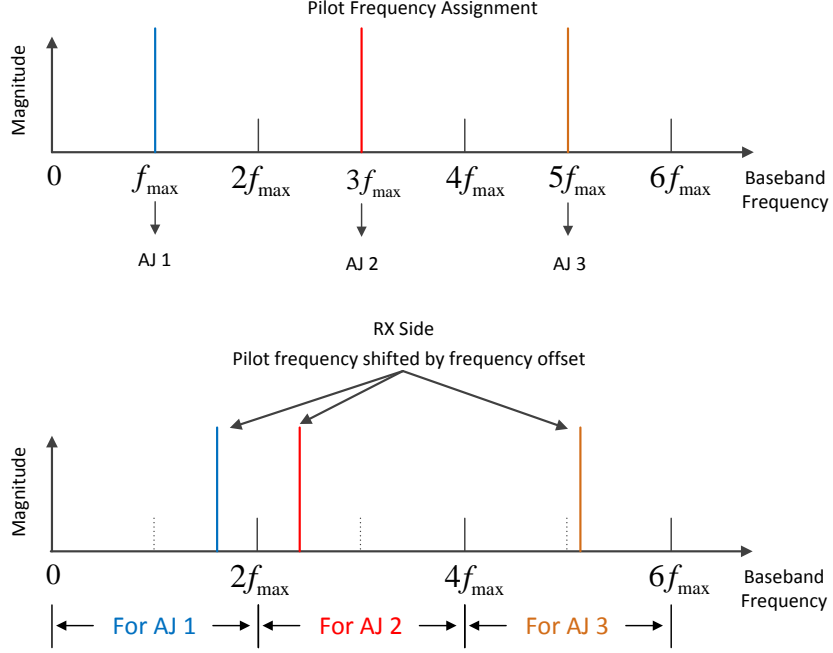


Figure 3.6: Pilot frequency assignment.

On the receiver side, for transmitted signal  $s_{i,m}$ , assuming the frequency offset is  $\Delta f_g$ , the authorized device will receive

$$\begin{aligned}
 y_{i,m} &= h e^{j\gamma} e^{j2\pi\Delta f_g t_{i,m}} s_{i,m} + n_{i,m} \\
 &= h e^{j\gamma} e^{j2\pi\Delta f_g t_{i,m}} (r_{i,m} + p f_m) + n_{i,m}.
 \end{aligned}$$

As  $r_{i,m}$  are pseudo-random samples, their energy is spread over a wide range of spectrum. On the other hand, the pilot frequency signals  $p f_m$  concentrate all their energy on a narrow band (1Hz wide), which will achieve a much larger magnitude, as shown in Figure 3.7. Therefore, on the receiver side, if the authorized device analyzes the spectrum of the received signals, it will find a spike within the designated shift range of the pilot frequency. Since the designated pilot frequency shift ranges of different ally jammers do not overlap, as shown in Figure 3.6, the pilot frequencies can be used for ally jammer identification.

Assuming the ally jammer  $AJ_g$  is identified, the authorized device knows its pilot frequency  $(2g-1)f_{max}$ . And as  $\Delta f_g + (2g-1)f_{max}$  has also been detected, the authorized device can infer their frequency offset  $\Delta f_g$ , which can be used further to compensate for their frequency offset.

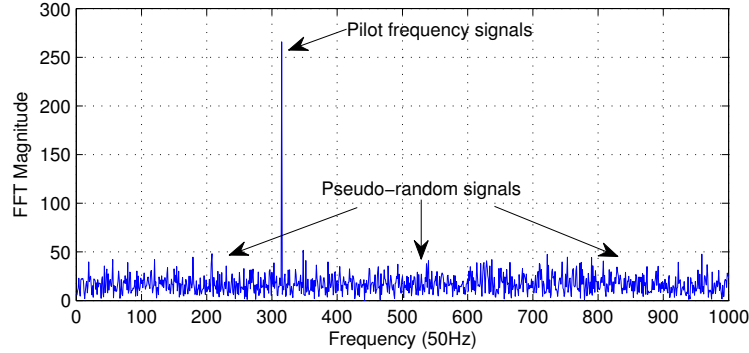


Figure 3.7: Received signal spectrum. Only show a portion of the whole spectrum.

### 3.3.4 Detecting and Recovering Transmissions

After synchronizing with the ally jamming signals, the authorized device needs to detect and recover potential transmissions from other authorized devices. Before a transmission is recovered, the authorized device cannot distinguish if it is authorized or unauthorized. Therefore, the authorized device will try to detect and recover all transmissions in the same way. For simplicity, in this section and the following section, we assume all transmissions are authorized transmissions. And we also assume that there is only one authorized transmission at one time, the media access control mechanism in ally friendly jamming will be presented later.

#### Re-synchronization & Transmission Detection

When the authorized device joins the network, it needs to synchronize with the ally jamming signals, this process is denoted as the *initial synchronization*. After initial synchronization, we have each authorized device re-synchronize with the ally jamming signals periodically. Figure 3.8 illustrates the re-synchronization process. Assuming that an authorized device re-synchronizes with the ally jamming signals every  $T$  time units. At the beginning of each re-synchronization period (e.g.,  $RS1$  in Figure 3.8), the authorized device compensates for the frequency offset, and correlates the received symbols with the regenerated ones to get the right alignment. Then it will estimate the channel by forming a quotient between each pair of received and transmitted (regenerated) jamming symbols. For example, as the frequency offset has already been compensated for and the noise is negligible, estimated channel coefficient for the samples in

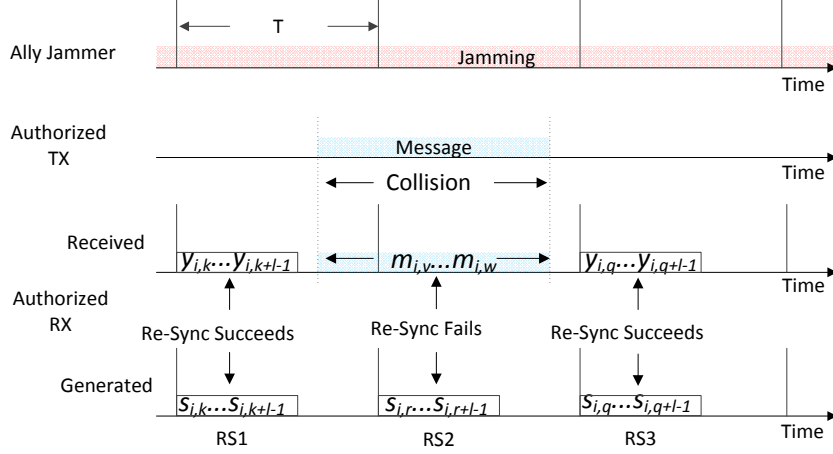


Figure 3.8: Transmission detection and recovery under ally friendly jamming. The authorized RX and the ally jammer are both in  $i$ -th epoch.  $s$  is the regenerated ally jamming signal,  $y$  is the received ally jamming signal,  $m$  is the received collided signal.  $T$  is the re-synchronization interval.

$RS1$  is

$$\begin{aligned}
 c_{i,u} &= \frac{y_{i,u}}{s_{i,u}} = \frac{he^{j\gamma} s_{i,u}}{s_{i,u}} \\
 &= he^{j\gamma}, u \in [k, \dots, k+l-1].
 \end{aligned}$$

If there are no transmissions other than the ally jamming signals in  $RS1$ ,  $c_{i,u}$  tends to be stable, as shown in Figure 3.9 (a). However, when there is an authorized transmission (e.g.,  $RS2$  in Figure 3.8), we have

$$c_{i,u} = \frac{he^{j\gamma} s_{i,u} + x_{i,u}}{s_{i,u}}, u \in [r, \dots, r+l-1],$$

where  $x_{i,u}$  is the received signal from the authorized transmission. The stableness of  $c_{i,u}$  is corrupted by  $x_{i,u}$ , as shown in Figure 3.9 (b). Thus by imposing a threshold on the standard deviation of the estimated channel coefficient, we can detect the existence of an authorized transmission under ally jamming.

To ensure that authorized device does not miss authorized transmissions, we set the re-synchronization interval  $T$  as a value smaller than the minimal packet transmission duration.

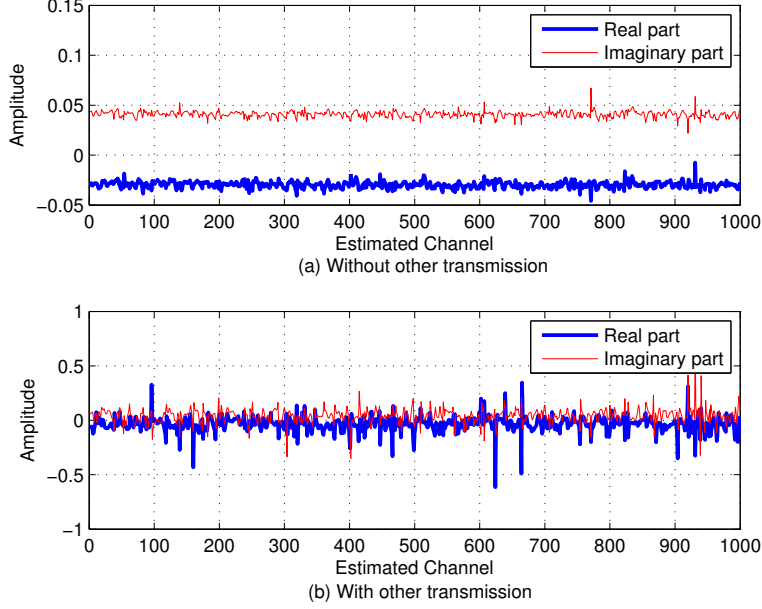


Figure 3.9: Estimated channel.

### Recovery of Authorized Transmissions

To remove the ally jamming signals, the authorized device firstly needs to estimate the corresponding components from the ally jammer in the received, mixed signals, then subtract them out to recover the detected transmissions.

Let us use the scenario shown in Figure 3.8 as an example, where the authorized device re-synchronizes successfully in  $RS1$ , but fails in  $RS2$  due to the collision. Since re-synchronization in  $RS1$  is successful, the authorized device can obtain the received ally jamming symbols in this interval (i.e.,  $y_{i,k}, \dots, y_{i,k+l-1}$  in Figure 3.8), which contain no strong interference (other strong signals, e.g., authorized transmission signals). As the frequency offset is already compensated for and the SNR is high, the least-square (LS) estimator can be employed to obtain a sufficiently accurate estimation of both  $h$  and  $\gamma$ .

The re-synchronization failure in  $RS2$  is caused by the collision of an authorized transmission with the ally jamming signals. Assuming the received signal components from the authorized transmission are  $x_{i,v}, \dots, x_{i,w}$ , the corresponding received ally jamming signal components in collision are  $y_{i,v}, \dots, y_{i,w}$ , then the received collided symbols  $m_{i,v}, \dots, m_{i,w}$ , are given by

$$m_{i,u} = y_{i,u} + x_{i,u} + n_{i,u}, u \in [v, \dots, w].$$

Assuming the estimated channel parameters are  $h'$  and  $\gamma'$ , the authorized device can get an estimation of  $y_{i,v}, \dots, y_{i,w}$ , say  $y'_{i,v}, \dots, y'_{i,w}$ , as

$$y'_{i,u} = h' e^{j\gamma'} \cdot s_{i,u}, u \in [v, \dots, w],$$

where  $s_{i,u}$  is the generated ally jamming symbol. Then the authorized transmission can be recovered by subtracting the estimated received ally jamming signals  $y'_{i,v}, \dots, y'_{i,w}$  from the received collided signals  $m_{i,v}, \dots, m_{i,w}$ . Thus, assuming the recovered authorized signal is  $x'_{i,u}$ , we have

$$\begin{aligned} x'_{i,u} &= m_{i,u} - y'_{i,u} \\ &= y_{i,u} + x_{i,u} + n_{i,u} - y'_{i,u} \\ &= h e^{j\gamma} \cdot s_{i,u} + x_{i,u} + n_{i,u} - h' e^{j\gamma'} \cdot s_{i,u} \\ &= (h e^{j\gamma} - h' e^{j\gamma'}) \cdot s_{i,u} + x_{i,u} + n_{i,u}, u \in [v, \dots, w]. \end{aligned}$$

As  $h'$  and  $\gamma'$  are accurate enough,  $(h e^{j\gamma} - h' e^{j\gamma'}) \cdot s_{i,u}$  is close to 0. Recall that the SNR of  $x_{i,u}$  is larger enough, then the recovered signal  $x'_{i,u}$  has sufficient SNR to be demodulated correctly, which further indicates the authorized transmission can be recovered readily.

Note that as the authorized device does not know the boundary of the authorized transmission, it will recover all the signals between two succeed re-synchronizations (i.e., all signals between *RS1* and *RS3* in Figure 3.8). Moreover, the authorized device can also use the received signals in the later successful re-synchronization interval to estimate the channel coefficients and recover transmission in previous intervals. For example, in the scenario shown in Figure 3.8, the authorized device can use  $y_{i,q}, \dots, y_{i,q+l-1}$  in *RS3* to estimate the channel, and recover the transmission in  $m_{i,v}, \dots, m_{i,w}$ .

### 3.3.5 Dealing with Multiple Ally Jammers

When an authorized device joins the system, it is likely that more than one ally jammers exist in the network, the authorized device needs to be able to remove the ally jamming signals from multiple ally jammers.

#### Synchronization with Multiple Ally Jammers

The authorized device can compute the spectrum of the received signals through FFT and identify all ally jammers by detecting all the spikes on the spectrum. It can further compensate for their frequency offsets and synchronize with each ally jammer through correlation.

Let us use an example to illustrate the process. Assuming that there are  $n$  active ally jam-

mers, from  $AJ_1$  to  $AJ_n$ , and the received signals at the authorized device are  $Y$ , which contain the jamming signals from all ally jammers. For one ally jammer, say  $AJ_g$ , if the authorized device does FFT on the received signals, it will find a spike within  $[(2g - 2) \cdot f_{max}, 2g \cdot f_{max})$ , which indicates that  $AJ_g$  is jamming the channel. And then the authorized device can compute their frequency offset  $\Delta f_g$  and find out  $AJ_g$ 's key  $k_g$  which can be used to generate the jamming signal sequences used by  $AJ_g$ , say  $s_g(1), s_g(2), \dots, s_g(n)$ .

Since the received signals  $Y$  contain the ally jamming signals from multiple ally jammers, we cannot compensate for  $AJ_g$ 's frequency offset on  $Y$  directly without disrupting other ally jammers' frequency offsets. To address this problem, the authorized device applies  $\Delta f_g$  on  $s_g(1), s_g(2), \dots, s_g(n)$  to mimic the same frequency offset effect. Then it can correlate the frequency offset compensated  $s_g(1), s_g(2), \dots, s_g(n)$  with  $Y$  to synchronize with the ally jammer  $AJ_g$ . Thus by finding out all the pilot frequency spikes on spectrum and repeating this process  $n$  times, the authorized device is able to synchronize with all ally jammers.

### Authorized Transmission Detection & Recovery

The detection of the authorized transmission under multiple ally jammers is similar to the detection under single ally jammer: when there is no authorized transmission, the estimated channels between these multiple ally jammers and the authorized device tend to be stable in short period (e.g., several milliseconds).

In the previous  $n$  active ally jammers example, the authorized device can get sample  $y(k)$  which contains ally jamming signals from all  $n$  ally jammers. As the frequency offsets have already been compensated for, we have

$$y(k) = \sum_{g=1}^n c_g \cdot s_g(k) + n_0(k), k \in [1, n],$$

where  $c_g = h_g e^{j\gamma_g}$  is the channel coefficient between the ally jammer  $AJ_g$  and the authorized device,  $s_g(k)$  is the jamming signal sent by the ally jammer  $AJ_g$  and  $n_0(k)$  is the white noise in received sample  $y(k)$ . Assuming  $\mathbf{y} = [y(1), y(2), \dots, y(n)]^T$ ,  $\mathbf{s}_g = [s_g(1), s_g(2), \dots, s_g(n)]^T$  and  $\mathbf{n}_0 = [n_0(1), n_0(2), \dots, n_0(n)]^T$ , we have

$$\mathbf{y} = [\mathbf{s}_1 \ \mathbf{s}_2 \ \dots \ \mathbf{s}_n] \cdot \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} + \mathbf{n}_0.$$

The distribution of the noise  $\mathbf{n}_0$  is known, and we know all the transmitted ally jamming

signals  $[\mathbf{s}_1 \ \mathbf{s}_2 \ \dots \ \mathbf{s}_n]$ . Thus the LS estimator can be used to solve the above equation and get the estimated channel coefficients

$$[c_1 \ c_2 \ \dots \ c_n]^T = (S^H S)^{-1} S^H \mathbf{y},$$

where  $S = [\mathbf{s}_1 \ \mathbf{s}_2 \ \dots \ \mathbf{s}_n]$ ,  $()^H$  denotes the conjugate transpose and  $()^{-1}$  is the matrix inverse operation. The authorized device can use different received signals to compute multiple versions of  $[c_1 \ c_2 \ \dots \ c_n]$  and further compute the standard deviation of each channel coefficient. If the mean value of all these standard deviations is larger than a threshold, then an authorized transmission is detected, the authorized device should start to remove the ally jamming signals.

By detecting the authorized transmission, the authorized device knows whether the received signals contain authorized transmission signals or not. Therefore, it can use the transmission-free samples to estimate the channel coefficients  $[c_1 \ c_2 \ \dots \ c_n]$ , then apply these channel coefficients to estimate the received ally jamming signals in the received collided signals and finally subtract them out to recover the detected transmission.

### 3.3.6 Dealing with Multiple Authorized Transmitters

In practice, it is possible that multiple authorized transmitters exist in the network. Since ally jamming signals will always occupy the channel, the traditional media access control (MAC) protocol (e.g., CSMA/CA) for wireless networking cannot be applied. It turns out that the transmission detection techniques can be used to solve this problem.

Before sending any packets, the authorized transmitter listens to the channel and computes the channel coefficients between itself and the multiple ally jammers by using the techniques described in Section 3.3.5. Suppose that there are  $n$  ally jammers and the computed channel coefficients are  $[c_1 \ c_2 \ \dots \ c_n]$ . If  $[c_1 \ c_2 \ \dots \ c_n]$  are stable for sometime (e.g., DIFS), then there is no other ongoing transmissions and the authorized transmitter will start to transmit, otherwise, it will back-off for some random time, listen to the channel and compute  $[c_1 \ c_2 \ \dots \ c_n]$  again.

## 3.4 Analysis

In this section, we provide an analysis of the proposed ally friendly jamming technique, including ally jamming power control and the limitation discussion.

Let us first clarify the notations. We denote the power of received ally jamming signals, the power of a received transmission (from either an authorized or unauthorized transmitter), and the power of received noise as  $J$ ,  $R$ , and  $N_0$ , respectively. The jamming to signal power ratio at the receiver side is  $JSR = \frac{J}{R}$ , the Signal to Noise Ratio is  $SNR = \frac{R}{N_0}$ . For simplicity, we

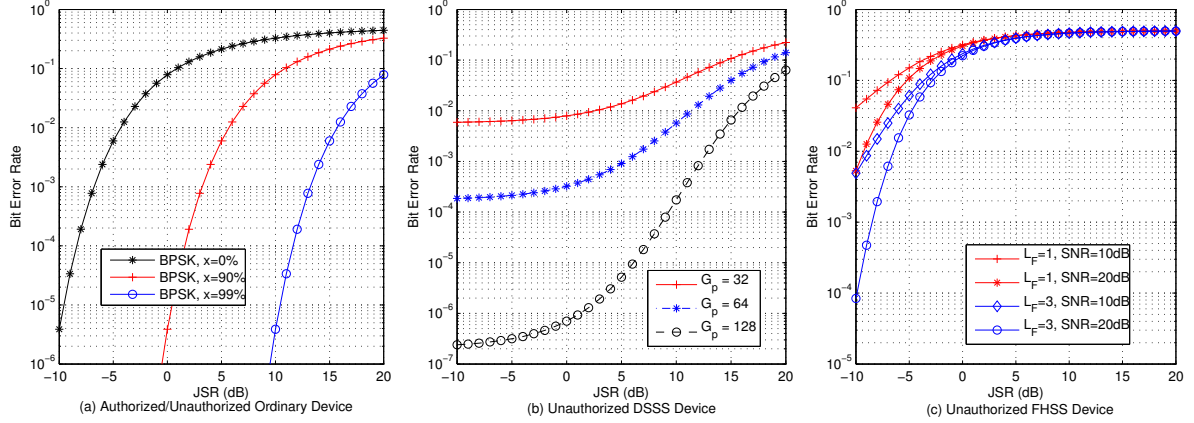


Figure 3.10: Bit error rate analysis.

assume authorized and unauthorized receivers observe the same received ally jamming powers and the same received transmission powers.

### 3.4.1 Maintaining Authorized Communication

We would like to understand how well the authorized communication can be maintained through analyzing the Bit Error Rate (BER) at authorized devices. According to [30], the BER of a wireless device is mainly dependent on its Signal to Interference and Noise Ratio (SINR) and the modulation method.

Let  $x$  be the portion of the ally jamming signal power that can be removed using our techniques. Consider the situation where the authorized devices use BPSK for modulation. Based on the result in [30], we can derive the BER as

$$P_e^a = Q \left\{ \sqrt{\frac{2}{\frac{1}{SNR} + JSR(1-x)}} \right\},$$

where  $Q(\cdot)$  is the Q-function (i.e.,  $Q(x)$  is the probability that a standard normal random variable will obtain a value larger than  $x$ ). Figure 3.10 (a) gives the BER values w.r.t.  $x$  and JSR, where  $\frac{1}{SNR}$  can be ignored as SNR is high enough. The results for other modulation methods can be derived similarly.

In our experiments, the percentage of removed jamming power  $x$  is between 99.2% and 99.6% (See Figure 3.14). It is generally agreed that wireless communication can be well maintained when the BER is less than  $10^{-3}$  [33]. This implies that we can maintain authorized wireless communication even if the JSR is as high as 17dB.

### 3.4.2 Disabling Unauthorized Communication

We consider three kinds of unauthorized devices: ordinary ones that do not use any anti-jamming techniques, those with DSSS-based anti-jamming capability, and those with FHSS-based anti-jamming capability.

#### Ordinary Unauthorized Devices

Unauthorized devices do not know the secret keys, and thus cannot regenerate the ally jamming symbols and remove them from the received signals. An ordinary unauthorized device may attempt to guess the jamming symbols to remove the jamming signals. Note that the random generation of the ally jamming symbols is essentially to randomly pick points from the constellation map. Even assuming a coarse-grained random generation with only 10 possibilities for the real and the imaginary parts of a random jamming symbol, there are  $10^2$  possible symbols in total. The probability of guessing  $y$  consecutive symbols right will be  $10^{-2y}$ , which quickly approaches 0 when  $y$  increases. Thus, the probability of removing the ally jamming signals through random guessing is very close to 0.

Based on the results in [30], if BPSK is used for modulation, the BER for an unauthorized device is

$$P_e^o = Q \left\{ \sqrt{\frac{2}{\frac{1}{SNR} + JSR}} \right\}.$$

The BER for other modulation methods can be derived similarly. Again assuming that the SNR is high enough,  $\frac{1}{SNR}$  can be ignored, we can get the BER as shown in Figure 3.10 (a), in which the line for  $x = 0\%$  shows the expected BER for an unauthorized device when BPSK is used for modulation. It is easy to see that when the jamming signal is  $10dB$  stronger than the power of a transmission, the BER of the unauthorized device is close to 50%, a value obtainable with random guesses, and their communication is disabled.

#### DSSS-based Unauthorized Devices

To jam DSSS-based unauthorized devices, the ally jammer needs to act as a broadband jammer [62] by increasing its symbol rate and injecting jamming signals with a bandwidth approximately the same as the DSSS signals from unauthorized devices. Assuming the spreading code length of unauthorized devices is  $G_p$  and BPSK is used for modulation, according to [62], we can estimate the BER of a DSSS-based unauthorized device under ally jamming as

$$P_e^d = Q \left\{ \sqrt{\frac{2G_p}{\frac{1}{SNR} + JSR}} \right\}.$$

Figure 3.10 (b) shows the BER when SNR=  $-10dB$ . It indicates that to disrupt the reception at an unauthorized receiver, the jamming signal must overcome the processing gain of spreading in DSSS. The result is consistent with the situation when ally friendly jamming is not used.

### FHSS-based Unauthorized Devices

To jam FHSS-based unauthorized devices, the ally jammer needs to use broadband jamming to make sure the jamming signals are strong enough on all hopping channels. Assuming a fast hopping system, the probability that the unauthorized device fails to receive the transmission in one hop is  $P_{e_k} = \frac{1}{2} \exp(-\frac{1}{2(\frac{1}{SNR} + JSR)})$ . According to [62], the BER of the FHSS communication under ally jamming is

$$P_e^f = 1 - \sum_{k=\lfloor \frac{L_F}{2} \rfloor + 1}^{L_F} \binom{L_F}{k} [P_{e_k}]^{L_F - k} (1 - P_{e_k})^k,$$

where  $L_F$  is the number of hops per data bit.

Figure 3.10 (c) illustrates the jamming performance against FHSS-based unauthorized devices. It is clear that when the JSR increases, the BER of FHSS-based unauthorized devices reaches 50% quickly and the communication is disabled.

### 3.4.3 JSR Trade-off

Maintaining authorized communication and disabling unauthorized communication have different requirements for JSR. JSR needs to be large to obtain effective jamming against unauthorized communication, but at the same time, JSR cannot be too large to affect authorized communication. Assuming that the BER of authorized devices should be at most  $P_e^{a,u}$ , and the BER of unauthorized devices should be at least  $P_e^{o,l}$  to disable their communication. Based on the earlier analysis, we can conclude that in order to maintain authorized communication and disable ordinary unauthorized devices, the JSR should be in the following range:

$$\left[ \left( \frac{2}{(Q^{-1}(P_e^{o,l}))^2} - \frac{1}{SNR} \right), \frac{1}{1-x} \left( \frac{2}{(Q^{-1}(P_e^{a,u}))^2} - \frac{1}{SNR} \right) \right].$$

For unauthorized devices using DSSS or FHSS, the jamming performance also depends on their processing gains besides JSR. When the processing gain is high enough, the ally jammer may not find a usable JSR to both allow authorized communication and disable unauthorized ones. However, authorized devices can also use anti-jamming techniques such as DSSS and FHSS. As a result, the JSR upper bound derived earlier can be significantly increased to allow effective jamming of unauthorized devices with anti-jamming capabilities.

### 3.4.4 Limitations

Ally friendly jamming provides us a desirable capability: disabling unauthorized wireless communication while still maintaining authorized wireless communication. This work may be viewed as the first step toward this goal. Several problems remain open for future works.

**Fast Identification of Ally Jammers:** Ally friendly jamming uses pilot frequencies for fast identification of ally jammers, which may introduce potential vulnerabilities. The attacker can inject or replay pilot frequency signals to mislead the authorized receiver’s synchronization process. Therefore, a more robust fast identification approach deserves further investigations.

**Fast Synchronization:** Shifting correlation based synchronization used by the authorized receiver is expensive in computation, and may have scalability issues, especially when the sample size and/or the number of ally jammers are large. Thus, a more computational efficient synchronization approach is desirable.

**Ally Friendly Jamming with MIMO Devices:** To make ally friendly jamming suitable for MIMO devices, we need to consider authorized/unauthorized MIMO devices (e.g., TX, RX) and MIMO ally jammers. One possible way of extending the current approach to the MIMO ally jammer case is: using a different key to generate jamming signals on each of the transmit paths of a MIMO ally jammer, and let the authorized receiver treat the MIMO ally jammer as multiple ally jammers. More studies are required for authorized/unauthorized MIMO devices cases.

**Handling Adversarial Jamming:** Authorized devices can use the anti-jamming techniques (e.g., DSSS and FHSS) to suppress the adversarial jamming signals after removing the ally jamming signals, which calls for efforts on the integration of ally friendly jamming and the anti-jamming techniques.

## 3.5 Implementation and Evaluation

We have implemented an “off-line processing” based prototype based on GNURadio and USRP. In the following of this section, we will give the implementation details and the corresponding evaluation results.

### 3.5.1 Experiment Setup

The prototype system consists of two ally jammers  $AJ_1$  and  $AJ_2$ , a transmitter, and a receiver. Each of them is implemented by a USRP N210 board connected to a laptop. Each USRP N210 uses a XCVR2450 daughter board operating in the 2.4GHz range as the RF front end. The receiver acts as an authorized device by using the techniques in ally friendly jamming to synchronize and remove the ally jamming signals, and as an unauthorized device by directly

demodulating the received signals. Our prototype implementation uses both GNURadio and MATLAB for signal processing. The USRP N210 uses a 2.5 PPM [7] temperature-compensated crystal oscillator (TCXO) as its frequency reference [9], the frequency drift is within  $[-6\text{KHz}, +6\text{KHz}]$  ( $2.4\text{GHz} \cdot 2.5 \text{ PPM} = 6\text{KHz}$ ). Therefore, the maximum frequency offset  $f_{max} = 12\text{KHz}$ , and the pilot frequencies for  $AJ_1$  and  $AJ_2$  are  $12\text{KHz}$  and  $36\text{KHz}$ , respectively.

The experiments contain three steps as described below. First, we use a PRNG with two different keys to generate the random floating point numbers with precision of 0.1 and uniformly distributed within  $[-1, 1]$ , which are then used to form the ally jamming symbols for  $AJ_1$  and  $AJ_2$  respectively.

Second, we keep the transmitter silent, turn on the receiver and let two ally jammers emit the ally jamming symbols simultaneously with the same transmit power. Ally jammers are about 2 meters away from the receiver. The ally jammer's symbol rate is  $5 \times 10^5$  sps (symbols per second). The receiver samples the channel at  $10^6$  sps and dumps the received samples in a file for the subsequent off-line processing. The samples collected in this step will be referred to as the *TX Off Samples*.

Third, we start the transmitter, which uses DBPSK modulation and sends packets with the length of 1,500 bytes at a data rate of 500kb/s. The interval between packets is 15ms. Ally jammers and the transmitter are about 2 meters away from the receiver and they all use the default transmit power with the same transmit gain. Ally jammers are still jamming the channel and the receiver still records the received samples in a file. The collected samples are termed as the *TX On Samples*.

### 3.5.2 Evaluation Methodology

The experimental evaluation consists of two parts: *micro-evaluation* and *macro-evaluation*. In micro-evaluation, we evaluate the performance of critical techniques used in ally friendly jamming. In macro-evaluation, we compare the bit error rates and packet loss rates for authorized and unauthorized devices under ally friendly jamming, including the case where unauthorized devices use DSSS for anti-jamming communication.

### 3.5.3 Micro-Evaluation

#### Synchronization

The authorized receiver does spectrum analysis on the *TX Off Samples* using FFT. Figure 3.11 shows the result on frequency domain when 10000 samples is used for FFT, from which we can clearly see that there is a spike at  $7.9\text{KHz}$ , and another one at  $32.7\text{KHz}$ . As  $7.9\text{KHz}$  is within  $[0, 24\text{KHz})$  and  $32.7\text{KHz}$  is within  $[24\text{KHz}, 48\text{KHz})$ , the authorized receiver knows that  $AJ_1$  and  $AJ_2$  are jamming the channel.

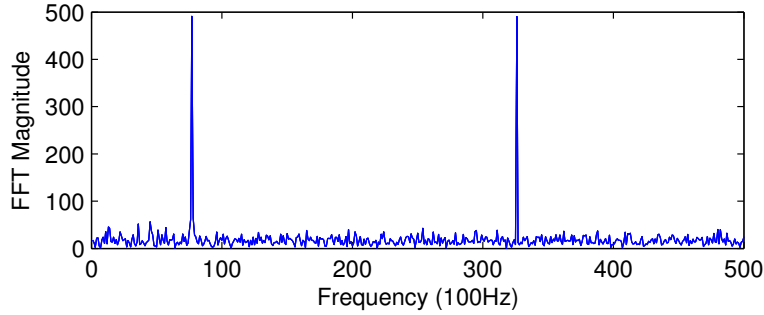


Figure 3.11: Identifying ally friend jammers.

After identifying ally jammers, the authorized receiver computes their frequency offsets, compensates for the frequency offsets on the locally generated symbols and correlates with the received jamming signals to synchronize with both  $AJ_1$  and  $AJ_2$ . As shown in Figure 3.12, there is a correlation peak for  $AJ_1$  at position 3190, which means that the timing offset between  $AJ_1$  and the authorized receiver is  $3190 \cdot T_p$ , where  $T_p$  is the sampling interval. The authorized receiver can use this offset to synchronize with the ally jammer  $AJ_1$ . Similarly, there is another correlation peak for  $AJ_2$  at position 22459. The authorized receiver can use the same process to synchronize with  $AJ_2$ .

We repeat this experiment 1,000 times with different samples. By using the correlation peak position as the indicator of timing offset, the success rate of synchronization is 100%. We also measure the time required for initial synchronization. It takes about 3 seconds for correlating  $10^6$  samples with a correlation length of  $10^3$  samples. After the initial synchronization, the re-synchronization takes less than 1 ms. Note that timing experiments are conducted on a laptop with an i7-2760QM CPU. The required time will be shorter on a dedicated radio chip. All of these experiments demonstrate that the authorized receiver can accurately synchronize with ally jammers.

### Detecting Transmissions under Ally Jamming

In this experiment, we examine how well the authorized transmission can be detected under ally jamming by using the *TX On Samples*. Since the packet length is 1500 bytes and the rate is 500kb/s, the packet transmission time is 24 ms. We set the re-synchronization interval as 10 ms. We adjust the transmit and receive gains such that the JSR is 5dB, 10dB, and 15dB, respectively, which are in the JSR trade-off range shown in Section 3.4.3. Then we examine the true positive and false positive rates of transmission detection for different thresholds on the standard deviation of the estimated channel coefficients. Figure 3.13 shows the result of the

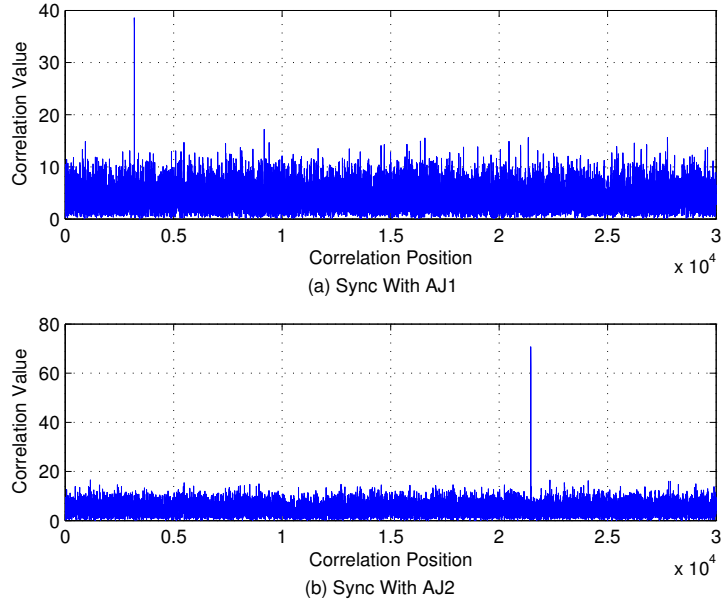


Figure 3.12: Synchronizing with multiple ally jammers. The correlation length is 1000 samples.

experiment. It is easy to see that there is a range of threshold values that allow the transmissions to be detected almost 100% with close-to-0 false positive rate. In other words, the detection of transmissions under ally jamming can be performed very precisely.

### Removal of Ally Jamming Signals

We want to know how well the authorized device can estimate and remove ally jamming signals when only ally jamming signals are received. We use the *TX Off Samples* collected when one and two ally jammers are on, respectively. After synchronization, we use the first 1000 samples to estimate the channel(s), predict the ally jamming signals in the following received samples, and then subtract them out from the received samples to check how much ally jamming power remains.

In our experiment, the percentage of jamming power removed by the authorized receiver depends on how many ally jamming samples we need to the estimate. Intuitively, as channel changes over time, if we apply the same estimated channel coefficients to estimate too many samples, the quality of estimation will degrade, and less jamming power will be removed. Figure 3.14 shows that the authorized device can remove 99.2% to 99.6% ally jamming power when the length of the estimated samples increases from 1,000 to 14,000. In other words, the vast majority of the ally jamming signal power can be effectively removed.

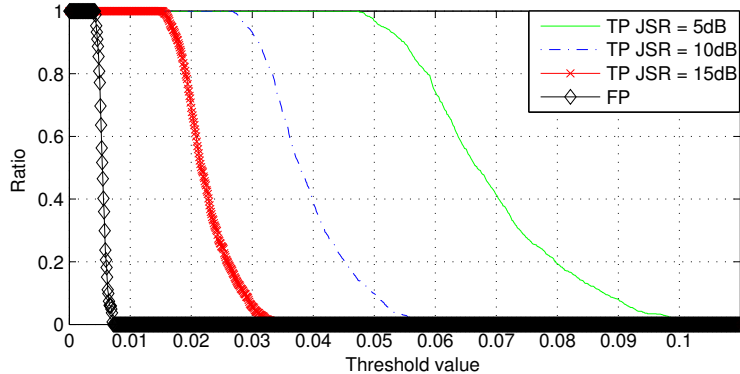


Figure 3.13: Transmission detection rate. FP is the false positive rate, TP is the true positive rate.

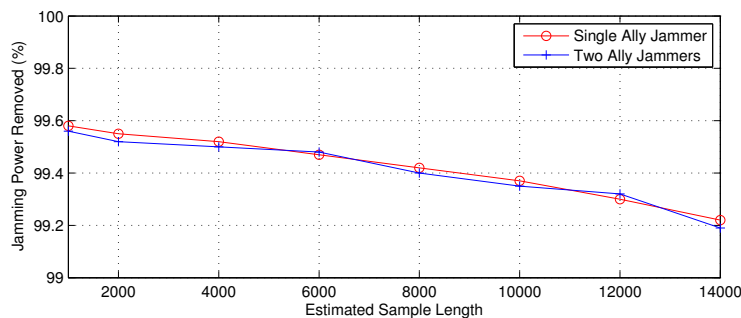


Figure 3.14: Removal of ally jamming signals.

### 3.5.4 Macro-Evaluation

The *TX On Samples* are used here. We adjust the transmitter's gain and ally jammers' gains to achieve different JSRs. The authorized receiver first detects the transmissions, recovers the transmitted signals, and then streams them into the demodulation blocks. In contrast, the unauthorized receiver demodulates the received samples directly.

Figure 3.15 (a) shows the BER for both authorized and unauthorized devices. It can be seen that as the JSR increases, the BER of the unauthorized receiver quickly increases to about 50%, a value achievable with random guesses. In contrast, with the ally jamming signals removal techniques, the authorized receiver can maintain close to 0 BER until the JSR exceeds 17dB. We use the GNURadio benchmark receiver to evaluate the overall packet loss rate. Figure 3.15 (b) shows the packet loss rates for both authorized and unauthorized receivers. Again, when the JSR increases, the packet loss rate at the unauthorized receiver quickly reaches 100%, while

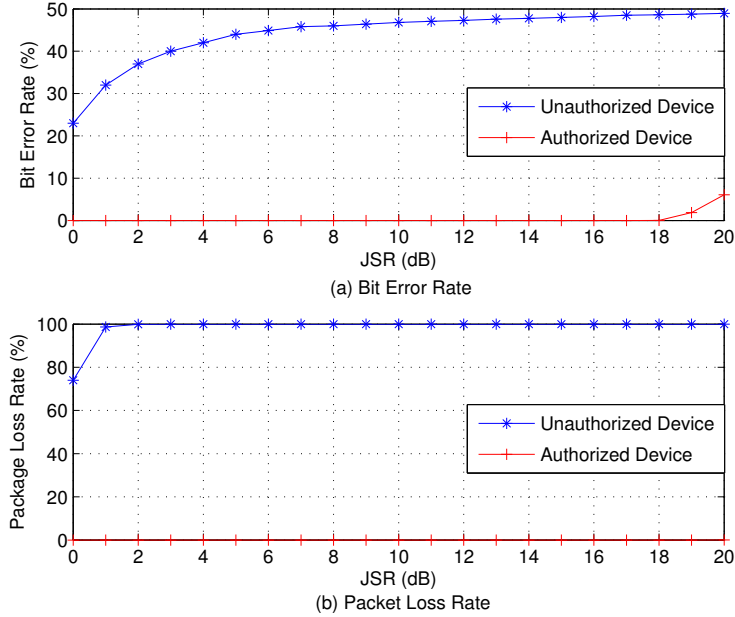


Figure 3.15: Macro-evaluation.

the packet loss rate at the authorized one remains close to 0 until the JSR reaches 16 dB. Unauthorized devices can certainly try to use Error Correction Code (ECC) to tolerate errors. However, with close to 50% BER, it is unlikely to reduce the packet loss rate much.

We also perform some preliminary evaluation of ally friendly jamming against unauthorized devices that are equipped with DSSS-based anti-jamming capability. In this experiment, we use IEEE 802.11b protocol running at 1 Mbps on unauthorized devices, which uses DSSS with an 11-bit Barker code for spreading and despreading [59]. More specifically, we use two laptops with 802.11b wireless adapters operating at the DSSS mode as unauthorized devices. We use another laptop connected to a USRP N210 board as the ally jammer. All these three devices are about 2 meters away from each other. We set the USRP using 2.452GHz frequency and the 802.11b wireless adapters using the same frequency (i.e., channel 9). We adjust the ally jammer's gain to make sure it has the same transmit power with the 802.11b transmitter. We test the packet loss rate at the 802.11b receiver side when different jamming symbol rates are used. (Note that higher symbol rates will cover wider spectrum.) Figure 3.16 shows that when the symbol rate for the ally jammer is more than  $600\text{ksps}$ , the communication between these 802.11b DSSS devices is disabled.

Note that though 802.11b DSSS mode is designed for wireless communication under interference, it is not intended as a strong anti-jamming solution. More in-depth evaluation is necessary

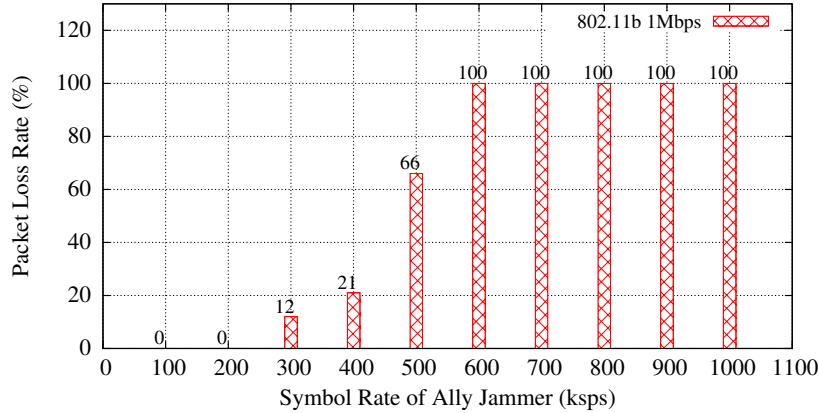


Figure 3.16: Jamming DSSS devices (ksps: kilo symbols per second).

to understand the performance of ally friendly jamming against powerful anti-jamming communication schemes.

### 3.6 Related Work

IMD Shield [32] is the most closely related work to ours. As discussed in the introduction, IMD Shield cannot achieve ally friendly jamming. We do not repeat it here.

This work is in general related to research on interference cancellation and suppression. Zigzag recursively applies interference cancellation to get the interference free signals from colliding ones [33]. Another Interference Alignment and Cancellation (IAC) technique was proposed to enable collaborative Access Points (APs) in MIMO LANs to decode more packets by controlling transmitted signals with proper vectors [34]. 802.11n<sup>+</sup> was proposed to use “antidote” signals to nullify the transmitted signals from other nodes in order to enable multiple access to wireless channels [46]. An implementation of successive interference cancellation (SIC) for ZigBee on software radios was presented in [36] which can decode concurrently transmitted packets. Moreover, SAM [81] provides a chain-decoding technique to decode concurrent frames. All these techniques assume regular modulated signals are transmitted and perform interference cancellation accordingly. Unfortunately, when the ally jamming signals mimic random noises, none of them can be used due to the challenges in synchronization and channel estimation.

Our proposed techniques have addressed these issues and advanced interference cancellation techniques to the next level.

Ally friendly jamming is also related to wireless jamming and anti-jamming research. For friendly jamming studies, Sankararaman et al. studied strategies of allocating friendly jammers to create wireless barriers which can prevent the eavesdropping [71]. There are also other literature (e.g., [54], [67], [91]) using friendly jamming to block the responses or unauthorized queries to protect particular wireless devices. For jamming and anti-jamming techniques, jamming attack models and several ways to detect jamming attacks have been studied in [92]. Game theoretical models have been developed for jamming and jamming defense [77, 95]. Spread spectrum techniques such as DSSS and FHSS have been traditionally used for anti-jamming wireless communication. In recent years, researchers have identified some weaknesses of such schemes due to shared keys and developed enhanced schemes, including Uncoordinated FHSS and its variations (e.g., [47, 78, 79, 80]), Uncoordinated DSSS and its variations (e.g., [48, 51, 64, 63]), and novel coding techniques (e.g., [15, 87]). Several filter designing jamming mitigation techniques have also been proposed [24, 25]. All these works are complementary to our results in this work.

## Chapter 4

# No Time to Demodulate - Fast Physical Layer Verification of Friendly Jamming

### 4.1 Preliminaries

Wireless communication aims to transfer information via radio frequency (RF) signals. The wireless transmitter in general modulates message bits into discrete base-band signals (signal symbols) first, then uses the digital to analog converter to convert these discrete signals to analog signals, and up converts them to radio frequency signals [30], and finally sends them out from its antenna.

Discrete base-band signals can be represented as complex numbers and the modulation process is equivalent to the mapping from bits to complex number points on the constellation diagram. A complex number can be represented in its polar form, i.e., a complex number  $a + bj = Me^{j\phi}$ , where  $M = \sqrt{a^2 + b^2}$  is its amplitude and  $\phi = \tan^{-1}\frac{b}{a}$  is its angle [53, 76].

RF signals travel through the wireless medium before being received by the receiver. The wireless channel will introduce the attenuation, phase shift, and additional noise to the original transmitted signals. As it is virtually impossible to operate two radios at exact the same frequency, the hardware of the transmitter and receiver will introduce a frequency offset [75]. Considering all these effects, for the transmitted signal  $x(i)$ , we have the received signal  $y(i)$  as

$$y(i) = he^{j\gamma}e^{j2\pi\Delta ft_i}x(i) + n(i),^1 \tag{4.1}$$

---

<sup>1</sup>The equation here is for single-tap channels.

where  $h$  is the channel attenuation,  $\gamma$  is the phase shift,  $\Delta f$  is the frequency offset,  $t_i$  is the sampling time and  $n(i)$  is the noise.

As the received signals are distorted by channel and hardware effects, the receiver needs to perform certain processes, such as frequency offset compensation and symbol synchronization, to demodulate these signals correctly. These processes not only complicate the receiver design, but also introduce certain delays to the reception process.

## 4.2 Assumptions and Threat Model

**Assumptions:** We assume that all ally devices, including friendly jammers, ally transmitters and ally receivers, are immobile. We also assume that friendly jammers and ally transmitters share a secret key which is unknown to unauthorized devices. As friendly jamming is normally applied for short range wireless communications, we assume that the wireless channels are single-tap and the propagation delay is negligible. We further assume that received signals have a sufficient signal to noise ratio (SNR) so that the friendly jammer can detect both the authorized and unauthorized wireless transmissions. We assume that clocks of all ally devices are loosely synchronized and the maximum clock drift is  $\Delta T$ . To facilitate the discussion, we assume that there are no adversarial jammers to authorized wireless communications.

**Threat Model:** The unauthorized devices will try various approaches to maintain their wireless communications under the friendly jamming. They may replay the received legitimate auth-preambles right before their transmissions so that their transmissions can bypass the auth-preamble verification at the friendly jammer. They may hijack the ongoing authorized transmissions by overshadowing the authorized transmission signals with much stronger unauthorized transmission signals right after the legitimate auth-preamble signals. The unauthorized devices may also try to remove the friendly jamming signals to maintain their wireless connections by exploiting advanced digital process techniques, such as the MIMO based attack approach [84].

## 4.3 Fast Friendly Jamming

### 4.3.1 Overview

The friendly jammer uses auth-preamble to distinguish authorized transmissions from unauthorized ones. Fig. 4.1 shows the overall design of fast friendly jamming. The ally transmitter prepends specially generated auth-preamble signals before its wireless transmission signals. The friendly jammer monitors channels and tries to verify received auth-preamble signals. Transmissions that are not accompanied by valid auth-preambles will fail the verification and be jammed by the friendly jammer. The friendly jammer plays two important roles in this process:

(1) it disables the wireless communications between unauthorized transmitters and unauthorized receivers, and (2) it prevents the ally receiver from accepting an unauthorized transmitter's signals.

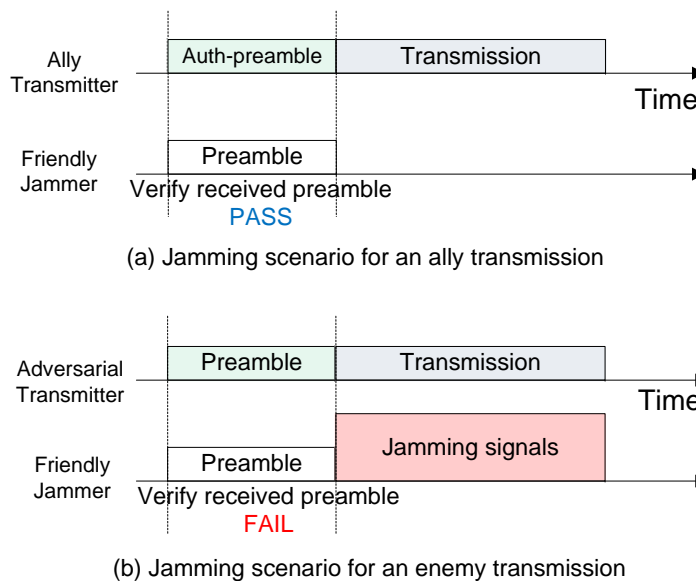


Figure 4.1: Jamming scenarios of fast friendly jamming.

The friendly jammer relies on auth-preambles to decide its action, the generation and verification of auth-preambles are crucial to fast friendly jamming. To realize fast friendly jamming, we propose to generate the auth-preamble signals by using the shared key and the time, and verify the received auth-preamble signals directly instead of demodulating auth-preamble signals into bits. In the following sections, we will give details of auth-preamble generation at the ally transmitter, and auth-preamble verification at the friendly jammer.

### 4.3.2 Auth-Preamble Generation

The auth-preamble signals need to be difficult for an adversary to predict, mimic and replay, while they should be easy for the friendly jammer to verify. As mentioned earlier, methods of using modulated bits as auth-preamble signals increase the reaction delay, as the friendly jammer needs to perform the time-consuming demodulation operations. Moreover, the modulated bits also give adversaries opportunities to mimic the auth-preamble signals due to the strong patterns in the modulated signals (e.g., phases). Therefore, we propose to use the shared key

and the timing information to control the generation of auth-preamble signal symbols directly.

As shown in Section 4.1, the auth-preamble signal symbols are discrete base-band signals which can be represented by complex numbers. Assuming the auth-preamble contains  $l$  signal symbols (complex numbers) and the symbol rate of the ally transmitter is  $r$  sps (symbol per second). Upon receiving an up-layer packet at time  $t_u$  (in seconds and is a float number), the ally transmitter first uses the shared key and  $\lfloor t_u \rfloor$  as the input to a pseudo-random number generator (PRNG) to generate  $2r$  floating numbers, denoted as  $a(0), a(1), \dots, a(2r - 1)$ . Then, floating numbers are used as the real and imaginary parts of  $r$  complex numbers, denoted as  $x(0), x(1), \dots, x(r - 1)$ . Each complex number  $x(i)$  can be formed by  $x(i) = a(2i) + a(2i + 1) \cdot j$ , where  $i = 0, 1, \dots, r - 1$ . Finally, the ally transmitter selects  $l$  consecutive complex numbers from the  $x$  sequence starting from the  $\lfloor f(t_u) \cdot r \rfloor$ -th symbol as auth-preamble signals, where  $f(t_u)$  is the fractional part of  $t_u$ .

The generated auth-preamble signals should be prepended before packet transmission signals. The auth-preamble and packet transmission signals need to be up-converted to RF signals before sending out from the antenna. Note that the final transmission may also contain a normal preamble after the auth-preamble for channel training or synchronization purpose.

### 4.3.3 Auth-Preamble Verification

#### Amplitude Differential based Correlation

The ally transmitter uses the generated pseudo-random complex numbers as auth-preamble signals, which brings challenges for the friendly jammer to verify the received copy of these signals. As mentioned earlier, the auth-preamble signals keep changing and resemble random noise, the channel and hardware effects are unknown. Therefore, traditional approaches cannot be applied for the verification of the auth-preamble signals.

To solve this problem, we propose *amplitude differential based correlation*, which enables the friendly jammer to verify received auth-preamble signals without demodulation. The basic idea is to use the *amplitude ratio* between two consecutive signals to tolerate the channel and hardware effects. For example, assuming that the transmitted auth-preamble signal is  $x(i)$  and the corresponding received one is  $y(i)$ . Observing that  $|e^{j\gamma} e^{j2\pi\Delta f t}| = 1$  and the received signals  $y$  have sufficient SNR, from (4.1), we have

$$\begin{aligned} |y(i)| &\approx |h e^{j\gamma} e^{j2\pi\Delta f t_i} x(i)| \\ &\approx |h x(i)|. \end{aligned}$$

Further observe that in slow fading environments, the change of channel attenuation  $h$  over a short period of time (e.g., a few milliseconds) is negligible. We denote the amplitude differential

value between two consecutively received signals  $y(i)$  and  $y(i + 1)$  as  $AD_{y(i)}$ . As the channel is single-tap, we have

$$\begin{aligned}
 AD_{y(i)} &= \left| \frac{y(i+1)}{y(i)} \right| \approx \left| \frac{hx(i+1)}{hx(i)} \right| \approx \left| \frac{x(i+1)}{x(i)} \right| \\
 &\approx AD_{x(i)}, i = 0, 1, \dots, l - 2.
 \end{aligned}
 \tag{4.2}$$

It is easy to see that the amplitude differential values between consecutive signals do not contain the channel and hardware effects (i.e., channel attenuation  $h$ , phase shift  $\gamma$ , and frequency offset  $\Delta f$ ), and thus the amplitude differential values of the received signals and the corresponding transmitted signals are roughly the same. Fig. 4.2 shows transmitted and received auth-preamble signals

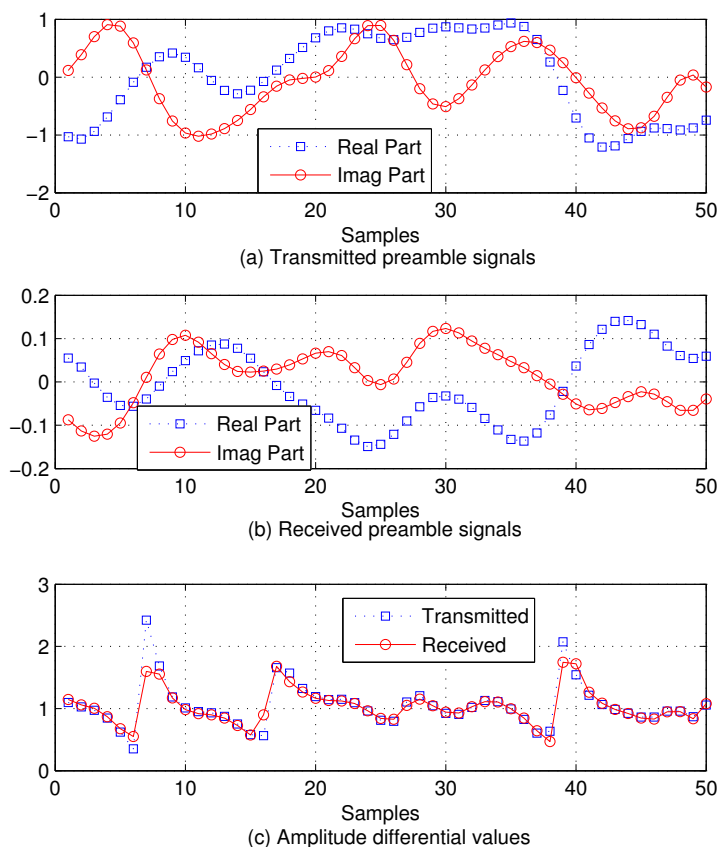


Figure 4.2: Transmitted auth-preamble signals, received auth-preamble signals and their amplitude differential values.

signals, and their amplitude differential values from our experiment based on GNURadio and USRP. We can see that due to channel and hardware effects, the received auth-preamble signals are very different from the transmitted ones, but their amplitude differential values are very close to each other.

In order to verify the received auth-preamble signals, the friendly jammer first uses the shared key and timing information to generate the auth-preamble signals locally and compute their amplitude differential values beforehand. Then, it computes  $AD_{y(0)}, AD_{y(1)}, \dots, AD_{y(l-2)}$  from the received auth-preamble signals  $y(0), y(1), \dots, y(l-1)$ . Finally, the friendly jammer correlates these two amplitude differential sequences to verify the auth-preamble signals. But before correlating, the friendly jammer needs to decide the correlation window for the locally generated signals.

Assuming that the friendly jammer received the auth-preamble signals at time  $t_a$ , the auth-preamble duration is  $T_d$ . To make sure that the transmitted auth-preamble falls into the correlation window of the locally generated signals, considering the clock drift  $\Delta T$ , the friendly jammer needs to generate the auth-preamble signals in the time window of  $[t_a - \Delta T, t_a + \Delta T + T_d]$ , denoted as  $g(0), g(1), \dots, g(m-1)$  and computes their amplitude differential values, denoted as  $AD_{g(0)}, AD_{g(1)}, \dots, AD_{g(m-2)}$ .

Both transmitted auth-preamble signals  $x$  and locally generated auth-preamble signals  $g$  are generated using the same key and  $g$  covers the time period of  $x$ . Thus,  $x$  should be a sub-sequence of  $g$ , which means that  $AD_y$  is a sub-sequence of  $AD_g$ . Therefore, to verify the auth-preamble signals, the friendly jammer does a shift correlation on  $AD_y$  with  $AD_g$ . Assuming the correlation result starts from  $AD_{g(i)}$  is  $\Gamma(i)$ , we have

$$\Gamma(i) = \sum_{z=0}^{l-2} AD_{g(i+z)} \cdot AD_{y(z)}, i = 0, 1, \dots, m-l. \quad (4.3)$$

The correlation result spikes when  $AD_y$  is aligned with  $AD_g$  correctly. To detect the spike, the friendly jammer imposes a threshold on the difference of the first and the second largest correlation outputs. If the difference is greater than the threshold, a spike is detected and the current transmission passes the verification, the friendly jammer will stay silent until the ongoing transmission finishes. If no spike is found during the correlation process, the verification fails and the current transmission is regarded as an unauthorized one, then the friendly jammer will start to jam the transmission.

### Efficient Correlation of Amplitude Differential Values

The correlation approach in (4.3) involves time consuming operations, such as float number multiplications and additions. To reduce the correlation time, we propose to use an approximate

but efficient method to perform the correlation between two sequences of amplitude differential values (i.e.,  $AD_y$  and  $AD_g$ ). Specifically, we transform each sequence of amplitude differential values into a bitmap by converting each amplitude differential value greater than a threshold to “1” bit (or “0” bit otherwise), as shown in Fig. 4.3. Given two equal-length bitmaps  $B_1$  and

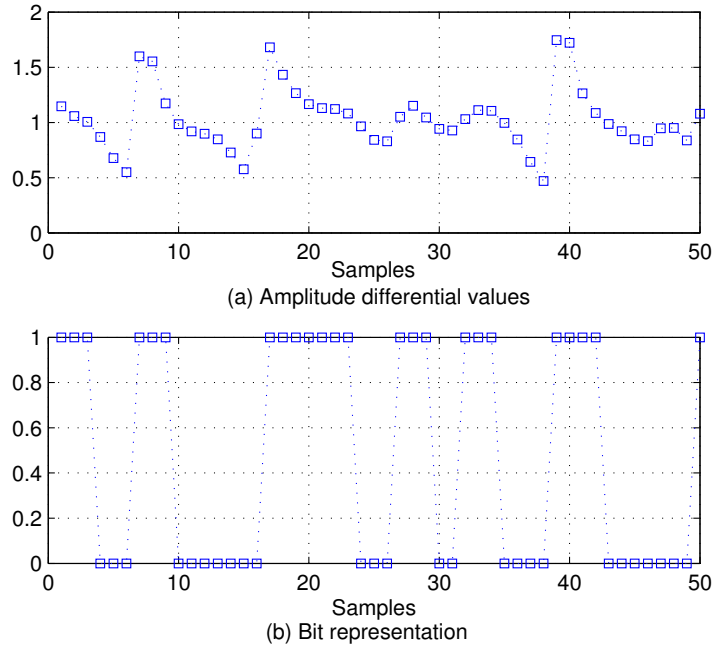


Figure 4.3: Amplitude differential values and their bit representation. The chosen threshold is 1.

$B_2$ , the correlation process in (4.3) can be expressed as  $\Gamma = |B_1 \wedge B_2|$ , where  $|B_1 \wedge B_2|$  is the weight (i.e., number of “1”) of bitmap  $B_1 \wedge B_2$ . Since the correlation between two bitmaps can be computed through bit-wise operations, this method can be executed efficiently.

## 4.4 Analysis

### 4.4.1 Against Different Kinds of Attacks

#### Relay Attack

In a relay attack, the adversary may relay and prepend the legitimate preamble signals before its own transmission signals to trick friendly jammers. The relay attacks can be classified into

two categories.

The first kind of attack may occur in a scenario with multiple friendly jammers (further discussed in Section 4.4.3). One friendly jammer  $AJ$  cannot receive signals from an ally transmitter  $AT$  as they are far away from each other. The adversary can relay  $AT$ 's preamble signals and append its own transmission signals to fool  $AJ$ . However, the adversary must receive all preamble signals before it can relay it. Thus, this kind of relay attack can be defeated by choosing the proper length for preambles. For example, assume that the preamble duration is  $T_d$ ,  $AT$  starts the preamble transmission at local time  $t_b$ , and the propagation delay for the relayed preamble is  $T'_p$ . When the relayed preamble arrives at  $AJ$ ,  $AT$ 's time is  $t_b + T_d + T'_p$ . Due to clock drift, at this time,  $AJ$ 's clock  $t_a$  is in  $[t_b + T_d + T'_p - \Delta T, t_b + T_d + T'_p + \Delta T]$ . The adversary hopes that  $AJ$ 's clock falls behind  $AT$ 's clock, so that it has more time to relay the preamble signals. Therefore, we consider the case  $t_a = t_b + T_d + T'_p - \Delta T$ , which benefits the adversary most. We know that the correlation window is  $[t_a - T_p - \Delta T, t_a + \Delta T + T_d]$ . To fail the verification,  $t_b$  needs to be out of the correlation window, which means

$$t_a - T_p - \Delta T > t_b.$$

$T_p$  is the maximum propagation delay that allows  $AJ$  to receive  $AT$ 's signals. In this case,  $AJ$  is far away from  $AT$  that it cannot receive signals from  $AT$ . Thus, the propagation delay for the relayed preamble signals  $T'_p$  should be greater than  $T_p$ , then we have

$$T_d > 2\Delta T.$$

When  $T_d > 2\Delta T$ , the start of the correlation window will be after time  $t_b$ . Then the friendly jammer cannot detect a validated preamble and will jam the unauthorized transmission.

Different from above, in the second kind of relay attack,  $AJ$  can receive  $AT$ 's signals. The adversary relays the preamble signals with a much stronger signal strength to create a capture effect on the authorized transmission at the  $AJ$  side. As the relayed preamble signals are legitimate ones, the friendly jammer may be tricked by just verifying the preamble. However, as the friendly jammer has the signal strength leash, it can detect the increase of received signal strength and will launch jamming against the unauthorized transmission.

## Replay Attack

The adversary may also launch the replay attack against fast friendly jamming. In the replay attack, the adversary records legitimate preamble signals and replays the recorded preamble signals right before its own transmission signals, as shown in Fig. 4.4. However, as the maximum propagation delay  $T_p$  is known, the replay attack can be thwarted if the authorized transmission

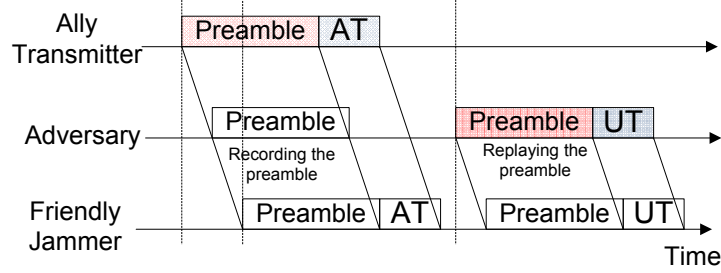


Figure 4.4: Replay attack scenario. AT is the ally transmission while UT is the unauthorized transmission.

duration (including the preamble)  $T_s$  achieves that

$$T_s > T_p + 2\Delta T.$$

The derivation is similar to the one in the relay attack discussion, so we don't repeat it.

### Preamble Hijack Attack

In this attack, the adversary will keep monitoring the channels. When it detects an authorized transmission, it sends its transmission signals at a carefully calculated time so that the unauthorized transmission signals append right after the legitimate preamble signals to “overwrite” the authorized transmission signals. The legitimate preamble signals are hijacked to trick the friendly jammer.

Again, similar to the second type of relay attacks, the friendly jammer can detect the increase of received signal strength due to the enforcement of the signal strength leash.

### MIMO Based Attack

Tippenhauer et al. proposed a MIMO based attack to remove the jamming signals from single friendly jammer to recover the confidential transmission signals protected by friendly jamming [84]. The same technique can also be used to eliminate the friendly jamming signals to recover the unauthorized transmission signals. To remove friendly jamming signals, the adversary's two antennas need to be placed at different locations which are equidistant to the friendly jammer. However, when multiple ( $> 2$ ) friendly jammers are deployed closely in the network, it is impossible to find such two locations that are the same distance away from all friendly jammers. Therefore, multiple jammers can be deployed to defeat this type of attack.

#### 4.4.2 Against Anti-Jamming Unauthorized Devices

The adversary may try to use anti-jamming techniques such as DSSS or FHSS to keep their wireless connections.

To defeat DSSS based unauthorized devices, the friendly jammer monitors channels. When detecting a low signal strength with an invalid preamble, the friendly jammer will jam channels as a high power jammer. In this work, we assume that unauthorized transmissions can be detected by the friendly jammer. However, when the spreading code is long, the signal power of the unauthorized transmission can be below the noise floor. In this case, the friendly jammer can apply anti-DSSS techniques such as inferring the spreading codes and jamming accordingly to overcome the spreading gain of the unauthorized communicators.

To defeat FHSS based unauthorized devices, the friendly jammer monitors channels. When it detects certain frequency hopping patterns, such as narrow-band and short-burst transmissions with invalid preambles, it will act as a broadband jammer to jam as many channels as possible.

#### 4.4.3 Impact of Multiple Friendly Jammers

In practice, the ally system may adopt multiple friendly jammers to enhance the jamming performance. For example, the friendly jammers can work together to defeat the earlier mentioned MIMO based attacks, to increase the jamming power against DSSS based unauthorized devices, and to jam more channels collaboratively than a single friendly jammer to defeat FHSS based unauthorized devices. Note that multiple friendly jammers can be easily deployed, because they do not interfere with the authorized transmissions.

#### 4.4.4 Communication Overhead

In fast friendly jamming, the auth-preamble signals introduce additional communication overhead. Assume that the auth-preamble has  $l$  symbols, the packet payload has  $n$  bytes, and  $b$  bits are modulated in one symbol, which means the payload has  $8n/b$  symbols. The overhead  $\psi$  is:

$$\psi = \frac{lb}{8n}.$$

For example, commercial Wi-Fi systems consist of short (64-bit) and long (128-bit) preambles specified in the 802.11 standard [40]. For a packet of 1500 bytes and a BPSK modulation, the communication overhead falls between 0.53% and 1.06%.

## 4.5 Experimental Evaluation

### 4.5.1 Experiment Setup

Our implementation is based GNURadio and N210 USRP. The prototype system contains a transmitter, a receiver, and a friendly jammer. Each node is a USRP connected to a host PC running GNURadio. We use XCVR2450 daughter boards operating on 2.4GHz as the RF front ends.

For the software parameter configuration, the transmitter generates pseudo-random float numbers with precision of 0.1 and uniformly distribution between  $[-1, 1]$ , then uses these floating numbers to form auth-preamble signals, as described in Section 4.3.2. The packet payload length is 1500 bytes. BPSK is used for payload modulation. The bit rate is  $250kbps$ , and sample per symbol is 4. Our implementation uses both GNURadio and MATLAB for signal processing.

In the evaluation, we will first measure the accuracy of amplitude differential based correlation, and then compare its execution time with the demodulation approach.

### 4.5.2 Auth-Preamble Verification Accuracy

In this part of experiments, we let the transmitter send auth-preamble signals, and the friendly jammer tries to verify the received auth-preamble signals using amplitude differential based correlation.

We repeat the experiment for 100 times. In each time, the transmitter transmits legitimate auth-preamble signals and the bogus auth-preamble signals (generated using a wrong key) with the modulated packet payload signals. The friendly jammer monitors the channel and computes the amplitude differential values for both the received auth-preamble signal samples and  $m$  interpolated locally generated auth-preamble signal samples. Considering the clock drift, we set the correlation window length for the locally generated auth-preamble signals as  $m = 10^4$ . Note that these signals can be generated beforehand to reduce the reaction time. The correlation can be transformed to bit-wise operations and executed efficiently. Assume that the computed amplitude differential values are denoted as  $AD_y$  and  $AD_g$ , respectively. If the difference of the first and the second largest correlation outputs is greater than a given threshold, the received auth-preamble signals are identified as legitimate auth-preamble signals.

In the experiments, the received auth-preamble lengths are 64 symbols (256 samples) and 128 symbols (512 samples). We evaluate the proposed techniques using the true positive rate (i.e., the rate that legitimate preambles are correctly identified) and false positive rate (i.e., the rate that bogus preambles are incorrectly identified as legitimate preambles). When using different thresholds, results for amplitude differential based correction and for the efficient variation (in Section 4.3.3) are shown in Fig. 4.5 and Fig. 4.6, respectively.

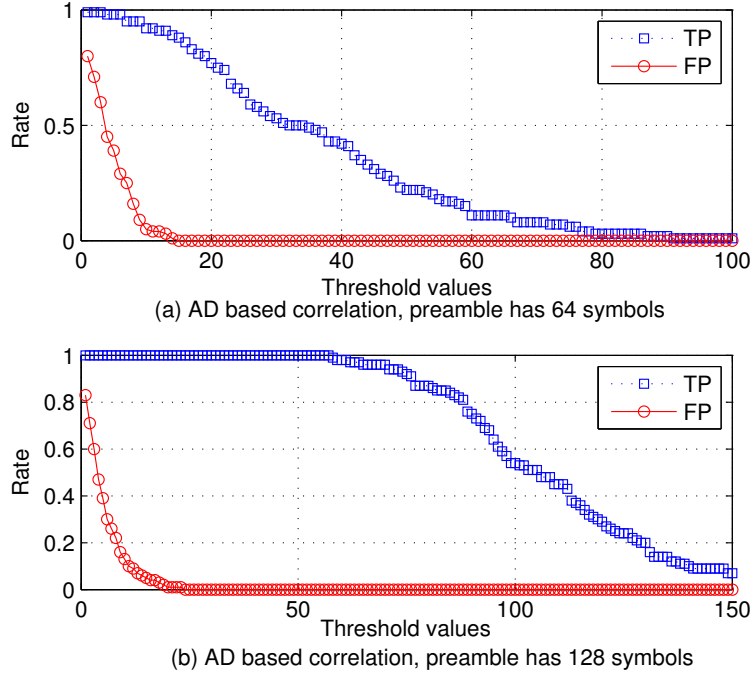


Figure 4.5: True positive and false positive of amplitude differential based correlation. TP is true positive and FT is false positive.

We can see that for both amplitude differential based correction and its efficient variation, when the auth-preamble has 128 symbols, there is a range of threshold values which achieve 100% true positive rate with 0% false positive rate. This means the amplitude differential based correlation can distinguish authorized and unauthorized transmissions accurately.

### 4.5.3 Execution Time

In this part of experiments, we want to compare the running time of efficient amplitude differential based correlation with the traditional demodulation approach. As the bit-wise operations are much faster compared to the complex float number operations, the dominating time-consuming factor for traditional demodulation approach is the demodulation operations; while for the proposed efficient amplitude differential based correlation approach, it is the computation of amplitude differential values.

To measure the demodulation time, we modify the benchmark receiver in GNURadio by connecting the receive path to demodulation related blocks (e.g., channel filter and demodulator) only and connecting the output directly to a null sink. Similarly, for counting amplitude differential value computation time, we connect the receive path to the amplitude differential

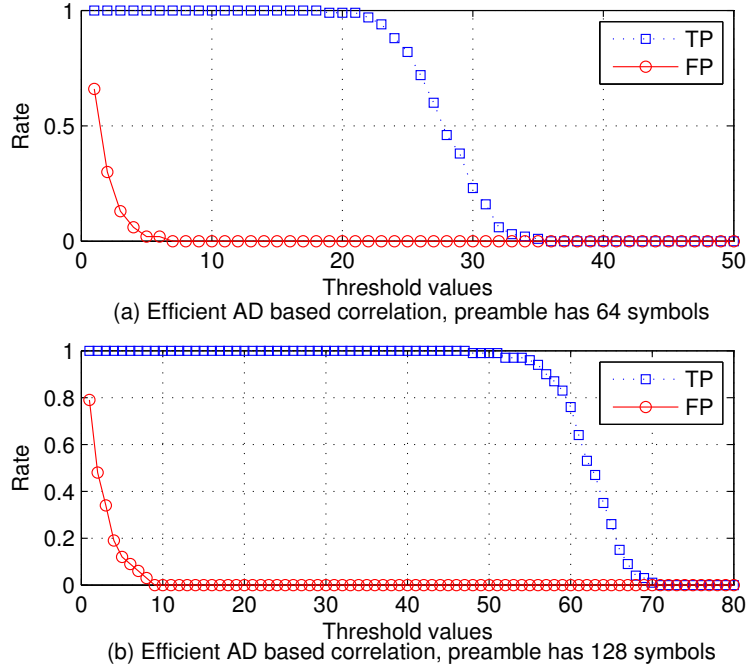


Figure 4.6: True positive and false positive of efficient amplitude differential based correlation.

values computation blocks and direct the output to a null sink. We measure the time of demodulating certain number of input signals and computing amplitude differential values for the same number of input signal samples.

When the number of input signals is small, the block setup time may dominate the real signal processing time. To make the results more accurate, we set the input signal length from  $2 \cdot 10^6$  to  $10^7$  and run each test for 100 times. We remove the greatest and the smallest ten execution times, the average execution time of remaining tests is shown in Fig. 4.7 .

It is easy to see that the computation of the amplitude differential values is in general 6-7 times faster than the demodulation operation. In other words, using efficient amplitude differential based approach rather than the demodulation approach, the friendly jammer can reduce reaction delay by 81.9% – 85.7%.

## 4.6 Related Work

The concept of friendly jamming has been recently explored to enhance the security of wireless communications (e.g., [67, 68, 91, 32]). For example, friendly jamming has been utilized to achieve information confidentiality and block unauthorized commands for RFID sys-

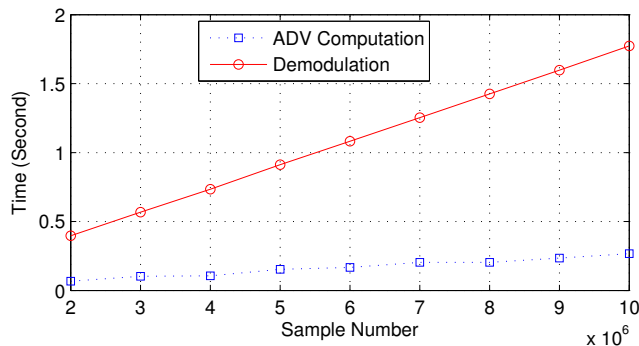


Figure 4.7: Time comparison. ADV is the amplitude differential values.

tems ([67, 68]) and implantable medical devices ([91, 32]). In Jamming for Good [54], the authors presented a reactive jamming scheme specifically designed for wireless sensor networks. A sensor node uses acceptable intervals such as the transmission frequency, to distinguish the malicious transmissions from benign ones. A reactive jammer against IEEE 802.15.4 network was implemented by Wilhelm et al. to demonstrate that the software-defined reactive jamming is feasible [85]. However, these schemes cannot be directly applied to fast friendly jamming.

However, all these works rely on the bit-level information to distinguish transmissions, which requires the demodulation of received signals and increases the reaction delay. While in Fast friendly jamming, the friendly jammer uses amplitude differential based correlation to verify the physical signals directly without demodulation.

RFID guardian [67] and [68] proposed to use friendly jamming to enforce the centralized access control policies to control the RFID access. IMDGuard [91] proposed an electrocardiogram based shared key establishment approach and a friendly jamming access control mechanism resilient to spoofing adversaries. IMDSHield [32] proposed to use friendly jamming to protect the un-encrypted wireless communications of implanted medical devices.

There are research works on the physical layer authentication techniques in recent years, such as hiding the authentication signals in the data signals [94] and using physical layer properties such as channel responses to authenticate users [89, 90]. These techniques are orthogonal to our work.

Our work is also related to anti-jamming techniques like DSSS (e.g., [51, 64]) and FHSS (e.g., [79, 80]), as well as general jamming studies. For example, Thuente et al. showed that compared with continuous jamming, intelligent jamming can achieve similar jamming effectiveness but cost less energy [83]. Xu et al. gave several jamming attack models and methods to detect jamming attacks [92]. Mobility of nodes was studied as a new approach for anti-jamming communication [38]. These works are complementary to ours. A reactive jammer

against IEEE 802.15.4 network was implemented by Wilhelm et al. to demonstrate that the software-defined reactive jamming is feasible [85]. Mobility of nodes is studied as a new approach for anti-jamming communication in [38]. Spread spectrum techniques such as DSSS and FHSS have been traditionally used for anti-jamming wireless communication. In recent years, several weaknesses introduced shared key establishment have been identified and the corresponding enhanced schemes have been developed, including Uncoordinated FHSS and its variations (e.g., [47, 78, 79, 80]), Uncoordinated DSSS and its variations (e.g., [48, 51, 64, 63]).

## Chapter 5

# Efficient In-band Wireless Pairing through Specialized CTS and Multi-carrier Communications

### 5.1 Preliminaries

In this section, we will give the background knowledge that will be referred in later sections, including the Wi-Fi Protected Setup and PBC, the 802.11 RTS/CTS, and the OFDM system.

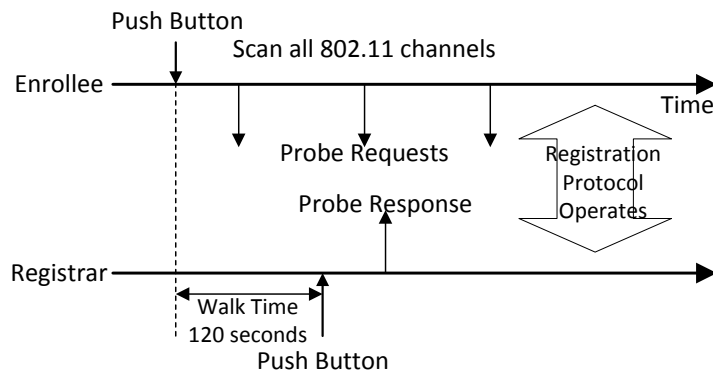


Figure 5.1: Push Button Configuration Pairing Process.

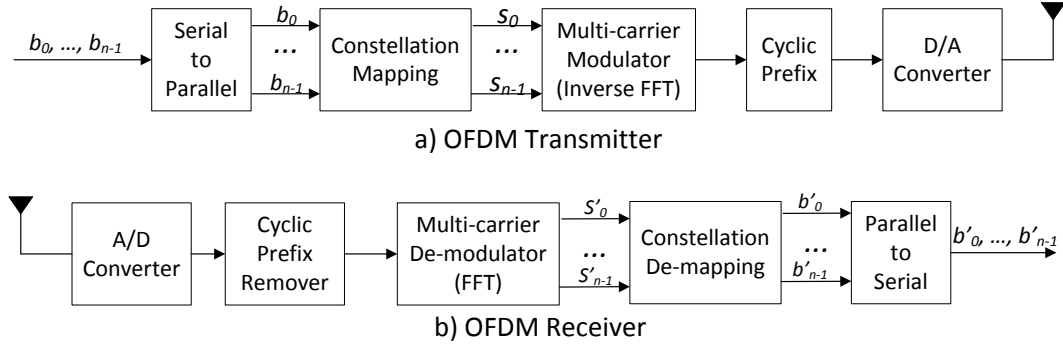


Figure 5.2: Basic structure of an OFDM communication system.  $b_i$  and  $b'_i$  are the message bits,  $s_i$  and  $s'_i$  are the discrete base-band signals.

### 5.1.1 Wi-Fi Protected Setup and PBC

Wi-Fi Protected Setup (WPS) was introduced by the Wi-Fi Alliance in 2006, aimed to allow users to add new devices to an existing network easily. It includes the PIN method, the push button method (a.k.a. push button configuration or PBC) and the out-of-band channel methods such as the near field communication method and the USB method. The WPS network usually contains two types of devices: the *registrar*, which has the authority to issue and revoke credentials in the network and the *enrollee*, which seeks to join the network. In typical WPS settings, the registrar may be integrated into a wireless access point.

To launch the pairing process, PBC requires a user to first push a button on the enrollee, then push the button on the registrar within a time interval of 120 seconds (i.e., the Walk Time) [12], as shown in Fig. 5.1. After its button being pushed, the enrollee will firstly scan all 802.11 channels by sending out its probe requests. If it receives responses from more than one registrar, it will abort its pairing attempt and signal a “session overlap” error to inform the user. For the registrar, the button press event causes it to check whether PBC probe requests have been received within the Walk Time. Similarly, if there are requests from more than one enrollee, it will also signal a “session overlap” error and abort the pairing process. If the enrollee only receives probe responses from one registrar and the registrar only receives probe requests from one enrollee, the enrollee and the registrar will proceed with the registration protocol operations which exchange messages containing the Diffie-Hellman primitives to establish the shared keys.

PBC protects against eavesdropping attacks. However, due to the absence of authentication to the pairing messages, PBC is vulnerable to the MITM attack.

### 5.1.2 802.11 and RTS/CTS

IEEE 802.11 standards use distributed coordination function (DCF) protocol to control the access to wireless physical medium. A wireless device must sense the wireless medium and wait for the medium to be idle for DCF Interframe Space (DIFS) duration before being permitted to transmit a data packet. Unlike the data packet transmission, the wireless device only needs to wait for the Short Interframe Space (SIFS) duration before it can send the acknowledgement packet of the received data packet. For 802.11 a/g implementation, the DIFS duration  $T_{difs}$  is 34  $\mu s$  and 28  $\mu s$ , respectively, while the SIFS duration  $T_{sifs}$  is 10  $\mu s$ .

802.11 uses a mechanism called RTS/CTS (Request to Send / Clear to Send) to reduce the collisions caused by hidden terminals [43]. The sender transmits a RTS that indicates how long it needs to occupy the channel. Upon receiving the RTS, the receiver replies a CTS with the expected channel occupation time. Any wireless devices that are close to the receiver can hear the CTS and they should hold their transmissions for CTS reserved duration. The devices that hear the RTS but not the CTS are free to transmit because they are far from the receiver and their transmissions will not interfere the receiver.

### 5.1.3 OFDM

Orthogonal frequency-division multiplexing (OFDM) is a frequency division multiplexing scheme which is used as a digital multi-carrier modulation method [6]. In OFDM systems, the wireless band is divided into multiple sub-carriers which are orthogonal to each other so that the cross-carrier interference is minimized. The data in OFDM is modulated and transmitted over these sub-carriers in parallel. Compared with the approaches of using single carrier, OFDM has advantages such as robustness against inter-symbol-interference (ISI) and fading caused by multi-path propagation [52] and high spectral efficiency, which make it a popular physical layer technology. The 802.11 a/g uses OFDM for physical layer modulation and the OFDM implementation has 52 sub-carriers, 48 of which are used for data transmission and 4 for equalization [73].

Fig. 5.2 shows a simplified basic structure of an OFDM communication system. The transmitter will first convert the bit sequence from serial to parallel so that different bits will be transmitted on different sub-carriers. Then these bits will be mapped to *discrete base-band signals (a.k.a., signal symbols)* on the constellation diagram [75, 76]. After that, the transmitter utilizes the inverse fast Fourier transform (IFFT) to convert the discrete signals to the signals occupying different sub-carriers, which is equivalent to multiplying these discrete base-band signals with different sub-carriers' frequencies. Then the *cyclic prefix* is added to the signal stream to serve as the guard interval and simplify the channel estimation and equalization. Finally, the transmitter converts the signals to analog signals and sends them out from its antenna.

The signals go through the wireless channel before being received by the receiver. The receiver does the reversed processing to get transmitted bits. It first converts the received analog signals to discrete digital signals, then removes the cyclic prefix and performs multi-carrier de-modulation by doing fast Fourier transform (FFT) on signals. Finally, it de-maps the signals to bits and converts the bits from parallel to serial to get the transmitted message.

## 5.2 Assumptions and Threat Model

**Assumptions:** We assume that all benign devices respect CTS requests. We further assume that the user launches the pairing process following the PBC standard [12] and buttons on the enrollee and the registrar are pushed within 120 seconds.

Moreover, we assume that the attacker cannot estimate the channels between the enrollee and the registrar; therefore, it cannot cancel the wireless transmissions between the enrollee and the registrar. Finally, we assume that the attacker cannot put the enrollee or the registrar into a Faraday cage to block all wireless signals.

**Threat model:** The attacker may launch MITM attacks by jamming benign pairing messages and faking its identity to pair with the enrollee and the registrar. The attacker has sufficient transmission power and the knowledge of the PBC and our proposed pairing protocols, so that it can align its transmissions precisely with benign pairing transmissions to make a capture effect that overshadows benign ones.

The threat model in this work is similar to the one in TEP [35]. The attacker may also adopt state-of-art communication techniques, such as the full-duplex radio to transmit and receive at the same time, or directional antennas so that only one pairing device can hear its jamming signals. The attacker can change its location and multiple attackers may collude with each other.

## 5.3 Protocol Design

To deal with MITM attacks, a secure pairing protocol needs to be able to prevent the attacker from hiding and tampering benign pairing messages without being noticed. As discussed earlier, the long channel occupation of the TEP pairing message may not only waste channel resources, but also interrupt wireless connections of nearby devices that operate on 2.4 GHz channels, including wireless medical devices.

To reduce the channel occupation time, we propose to use OFDM sub-carriers in the frequency domain as the on-off slots, to make the pairing message *tampering-proof*. We further propose to use a specially crafted CTS request and the cooperation between the pairing devices to achieve pairing message *hiding-proof*. In the following of this section, we will give details

of these two techniques. For ease of reference, we summarize all terms used in the following sections in Table 5.1.

Table 5.1: Terminologies

| Term       | Definition                                           |
|------------|------------------------------------------------------|
| $MSG_e$    | Pairing probe request from the enrollee              |
| $MSG_r$    | Pairing probe response from the registrar            |
| $FMSG_e$   | Fake enrollee request from the attacker              |
| $FMSG_r$   | Fake registrar response from the attacker            |
| $T_{sifs}$ | SIFS duration                                        |
| $T_{difs}$ | DIFS duration                                        |
| $T_m$      | Benign pairing message duration                      |
| $T_s$      | CTS reserved duration                                |
| $T_w$      | The maximum waiting time between two pairing queries |

### 5.3.1 Tampering-Proof Pairing Message

To protect the message integrity as well as to avoid using the on-off slots in the time domain, we propose to convey the hash digest of the pairing message using on-off sub-carriers in the frequency domain.

The pairing message payload contains the Diffie-Hellman public key. Upon receiving the payload bits, the transmitter will first generate a cryptographic hash digest for them, then the transmitter maps the bits of the hash digest to the sub-carriers in the frequency domain, as shown in Fig. 5.3.

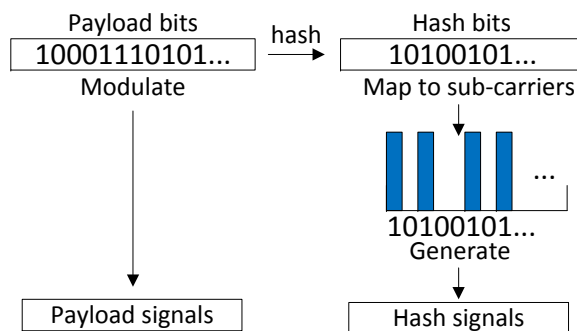


Figure 5.3: Packet signal generation at the transmitter.

For simplicity, we assume the number of sub-carriers is equal to the number of hash bits. The mapping between a hash bit and a sub-carrier is one-to-one mapping, which means the  $i$ -th bit is mapped to the  $i$ -th sub-carrier. If the hash bit is 1, the transmitter generates signals to occupy the mapped sub-carrier, otherwise, it keeps that sub-carrier idle. The occupied sub-carrier is called the *on sub-carrier* while the idle sub-carrier is the *off sub-carrier*. This processing step makes sure that the on sub-carrier has the transmission energy while the off sub-carrier does not, so that the receiver can apply the energy detection on a sub-carrier to decode its carried hash bit. Finally the transmitter will append the hash signals to the modulated baseband payload signals, up-convert them to RF signals, and send these signals out from its antenna.

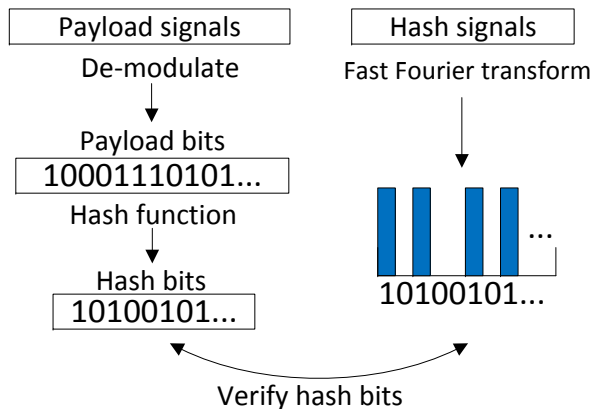


Figure 5.4: Hash bits verification at the receiver.

At the receiver, it firstly de-modulates the received payload signals to bits and computes the cryptographic hash digest, say  $H_c$ . Then, it does a fast Fourier transform (FFT) on the received hash signals and decodes the carried bit to 1 if the energy on that sub-carrier is greater than a certain threshold or to 0 otherwise, to get the hash digest  $H_h$ , as shown in Fig. 5.4. Finally, the receiver will verify  $H_c$  against  $H_h$ . If they match, the receiver will accept this pairing message; otherwise, the pairing message will be treated as a collision as it may be tampered by the attacker.

The last step to achieve pairing message tampering-proof is to balance the 1 and 0 bits. As the attacker cannot cancel the energy of the on sub-carriers, the receiver can detect any tampering on the benign pairing message if the numbers of on sub-carriers and off sub-carriers are equal. In this work, we use the encoding scheme in TEP [35], which takes  $N$  input bits ( $N$  is an even number) and produces 1 and 0 balanced bits with a length of  $N + 2\lceil \log N \rceil$ .

Note that in the original OFDM design, sub-carriers are used to transmit signals. It is

these signals that can be decoded to bits. But in our design, *signals themselves do not carry any information, but the energy presence or absence on the sub-carrier conveys the hash bit.* Therefore, when working with 802.11 a/g OFDM, the receiver doesn't need to reserve 4 sub-carriers for equalization, all 52 sub-carriers can be used for hash bit mapping.

The cryptographic hash algorithms, such as MD5, SHA-1 and SHA-256, produces 128, 160, 256 bits hash digests, respectively, which cannot be one-to-one mapped to the 52 sub-carriers of 802.11 a/g at one time. To address this problem, the transmitter can divide the hash bits into multiple 52-bit groups to fit the sub-carriers. The last group can be padded with "01" to 52 bits. Each group contains 52 bits which can be one-to-one mapped to the 52 sub-carriers at one time. The pairing devices should have an agreement on the size of the hash digest, so that any attempt to append fake hash signals after the benign transmission can be detected by the receiver.

### 5.3.2 Hiding-Proof Pairing Message

The basic idea of using a specially crafted CTS request and the cooperation between the pairing devices to make the pairing message hiding-proof is that benign devices will respect the CTS packet and remain silent during the CTS reserved period. On the contrary, the attacker has to jam the pairing packet even though it receives the CTS packet. Otherwise, the pairing device will receive the benign pairing message which contains the Diffie-Hellman public keys, and the attacker has no chance to launch the MITM attack successfully. Therefore, the pairing devices can use the collisions happened in the CTS reserved duration as an indicator of existence of the attacker in the network. In other words, if the enrollee sends out the CTS successfully and detects collisions in the CTS reserved duration, it knows that the attacker exists in the network, thus it will abort the pairing process and signal a session overlap error to the user. In this way, the enrollee will not be tricked to pair with the attacker.

When no attacker exists in the network, the interactions between the enrollee and the registrar are shown in Fig. 5.5. The  $MSG_e$  and  $MSG_r$  are the pairing messages from the enrollee and the registrar, respectively. The enrollee firstly transmits probe request  $MSG_e$  which contains a special CTS to reserve a time slot  $T_s$ . To ensure that even though the attacker messes up the timing of the registrar, the enrollee still can receive the registrar's reply or the collision during the CTS reserved duration (details are included in Section 5.3.6), we set the CTS reserved duration as two pairing messages duration and the guard interval (SIFS duration). Assuming the pairing message duration is  $T_m$  and the SIFS duration is  $T_{sifs}$ , then the CTS reserved duration  $T_s = 2 \cdot T_m + 2 \cdot T_{sifs}$ .

$MSG_e$  contains the Diffie-Hellman public keys as the payload, the hash signals and a CTS, which are separated by a guard time of SIFS duration. The hash signals are used against

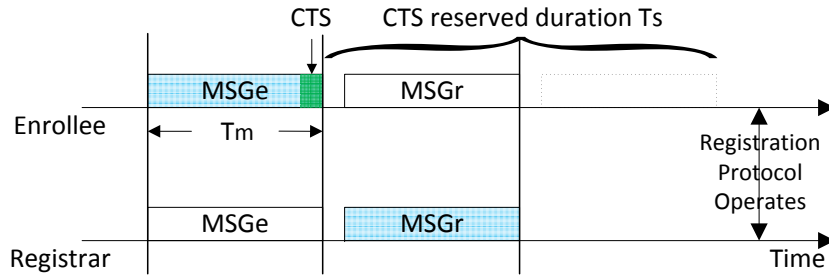


Figure 5.5: Pairing scenario with no attackers. The buttons on the enrollee and the registrar were pushed.

tampering attacks, which is detailed in Section 5.3.1. The format of  $MSG_r$  is the same with  $MSG_e$  but does not contain CTS.

Upon receiving the CTS from the enrollee, the nearby 802.11 devices will hold their transmissions for the CTS reserved duration. For the registrar, after it receives the pairing message which contains the special CTS from the enrollee, it will transmit its response  $MSG_r$  to the enrollee. Because there is no attacker and  $MSG_r$  is sent during the CTS reserved duration, the enrollee will receive a collision-free  $MSG_r$ . After receiving  $MSG_r$  and  $MSG_e$  respectively, the enrollee and the registrar can use the Diffie-Hellman public keys to establish a shared key, which can be further used to protect their wireless communications for the following pairing operations.

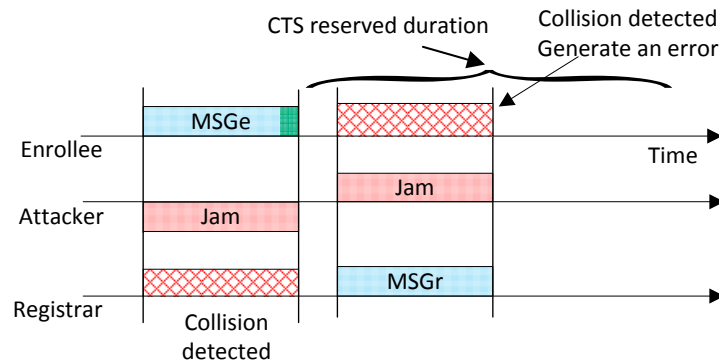


Figure 5.6: Pairing scenario with jamming attacks. The buttons on the enrollee and the registrar were pushed. The attacker uses directional antenna, so that the jamming signals below the time line can only be received by the registrar while the ones above time line can only be received by the enrollee.

When an attacker exists in the network, it has to jam  $MSG_e$  and  $MSG_r$  to disrupt the pairing process. As is shown in Fig. 5.6, the attacker jams the probe request  $MSG_e$  to form a collision at the registrar. The directional antenna can be used, so that only the registrar hears the jamming signals. As a result, the probe request  $MSG_e$  is *hidden* by the attacker, and then the registrar will not notice the pairing attempt from the enrollee.

To prevent the attacker from hiding the pairing message, after its button being pushed, whenever the registrar detects a collision which is greater than or equal to the duration of a benign pairing message duration  $T_m$ , it waits for SIFS time and transmits its response  $MSG_r$  no matter whether the channel is occupied or not. In the meantime, it regards long collisions as the possible attacks, and aborts the pairing process when long collisions are heard.

For the response  $MSG_r$ , the attacker has to jam it again (otherwise the public key in  $MSG_r$  can be received by the enrollee and the MITM attack will fail). As a result, the enrollee will detect a collision during the CTS reserved duration. As the benign devices respect CTS, the enrollee deduces that the collision is caused by the attacker's jamming, it will generate a session overlap error and abort the pairing process.

Note that the walk time between two button pushes is 120 seconds (as shown in Fig. 5.1). When the enrollee transmits the  $MSG_e$ , it is possible that the button on the registrar has not been pushed yet and the registrar will not reply to the  $MSG_e$  or the collision. This gives the attacker a chance to pair with the enrollee first. To prevent this kind of attack, both the enrollee and the registrar will scan the channel for  $120\text{ sec} + \#channel \times (T_m + T_s + T_w)$ , before entering the next pairing stage. Moreover, the attacker may jam the enrollee all the time to give it a wrong impression that the channel is busy, and try to pair with the registrar at the same time. To defeat this kind of attacks, the enrollee will wait for maximum  $T_w$  seconds, then it will transmit  $MSG_e$  even though the channel is busy.

### 5.3.3 Distinguishing Attack Collisions from Normal Ones

Collisions are common in the wireless environment. If reacting to every collision it detects, the registrar will abort pairing process due to high false positive rate. Moreover, the probe response  $MSG_r$  will be sent many time (equals the number of collisions), which will introduce excessive channel occupation.

To address this problem, we propose to enlarge the duration of the pairing message. Most of collisions in 802.11 network involves two packet transmissions, and the collision duration is shorter than two packet transmissions. The maximum packet size used by the upper layer in 802.11 is typically 1,500 bytes. Therefore, we can pad the pairing message to be 3,000 bytes, so that its duration will be about twice of a 1,500-byte packet transmission duration. Whenever the registrar detects a collision that is equal to or longer than two packet transmissions, it

deduces that the collision is very likely caused by the jamming attack. Then the registrar will send its pairing message  $MSG_r$ . Because collisions that involves three or more transmissions are much less likely than the collisions caused by two transmissions in a 802.11 network, this design can reduce most of the channel occupation caused by unnecessary responses from the registrar.

### 5.3.4 Reducing Pairing Message Collisions

In order to let the pairing device distinguish the collisions caused by jamming from normal collisions, the pairing message is padded to be longer than before. The long pairing message may increase collision chances at the sender side due to the hidden nodes or other reasons.

To address this problem, we propose to change the format of the pairing message. Instead of putting the CTS at the end of a pairing message, the sender (the enrollee or the registrar) can put the CTS at the beginning of a pairing message, before the payload signals, as shown in Fig. 5.7.

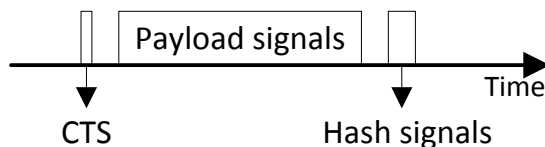


Figure 5.7: Pairing message format. The CTS, the payload signals and the hash signals are separated by a SIFS.

The sender first transmits the CTS request. If the CTS is sent out without detecting any collisions, it will then send the long payload signals and the hash signals. As the CTS package is much shorter than the normal packet, the collision chance on CTS package will be very small.

Note that the CTS reserved durations are different for the  $MSG_e$  and  $MSG_r$ . For  $MSG_e$ , the enrollee needs to reserve the time for an initial  $MSG_e$  and a time period for hearing the reply, then the reserved time should be  $T_m + T_{sifs} + T_s$ . On the other side, the registrar only needs to reserve the channel for the response  $MSG_r$ , whose duration is  $T_m$ .

### 5.3.5 Integrating with PBC

With the hiding-proof and tampering-proof pairing message, we can form the protocol for PBC. For the enrollee, after the button is pushed, it will keep scan all channels. To make sure that

the registrar can receive at least one pairing request from the enrollee, we set the timer as 120 sec + #channels $\times$ ( $T_m+T_s+T_w$ ). The enrollee uses  $cnt_r$  to count the new  $MSG_r$  number and uses a flag  $collision$  to indicate whether a collision happens during the CTS reserved time. Its protocol is as follows. Here  $T_s$  is the CTS reserved duration. As the benign devices remain silent

---

```

function ENROLLEE-RUN()
   $cnt_r \leftarrow 0$ ,  $timer \leftarrow 0$ ,  $collision \leftarrow false$ 
  while  $timer \leq 120 \text{ sec} + \# \text{ channels} \times (T_m + T_s + T_w)$  do
    move to next channel and send  $MSG_e$ 
    if hear a new  $MSG_r$  in  $T_s$  then
       $cnt_r \leftarrow cnt_r + 1$ 
    else if hear a collision in  $T_s$  then
       $collision \leftarrow true$ 
    end if
    update  $timer$ 
  end while
end function

```

---

during  $T_s$ , the collision happens due to the attacks to hide or tamper  $MSG_r$ .

Similarly, the registrar uses  $cnt_e$  to count all pairing request from different enrollees. After its button being pushed, whenever the registrar hears a new  $MSG_e$  or a collision that is longer than  $T_m$ , it will send a  $MSG_r$  to claim existence of the pairing message. Note that after the timer expires, the enrollee (the registrar) will continue the pairing process if and only if  $cnt_r$  ( $cnt_e$ ) is 1 and  $collision$  is false. Otherwise, they will raise an error and try to pair later.

---

```

function REGISTRAR-RUN()
   $cnt_e \leftarrow 0$ ,  $timer \leftarrow 0$ ,  $collision \leftarrow false$ 
  while  $timer \leq 120 \text{ sec} + \# \text{ channels} \times (T_m + T_s + T_w)$  do
    if receive a  $MSG_e$  from a new enrollee then
       $cnt_e \leftarrow cnt_e + 1$ 
      reply a  $MSG_r$ 
    else if hear collisions longer than  $T_m$  then
       $collision \leftarrow true$ 
      for any length of  $T_m$ , reply a  $MSG_r$ 
    end if
    update  $timer$ 
  end while
end function

```

---

### 5.3.6 Example Attack Scenarios

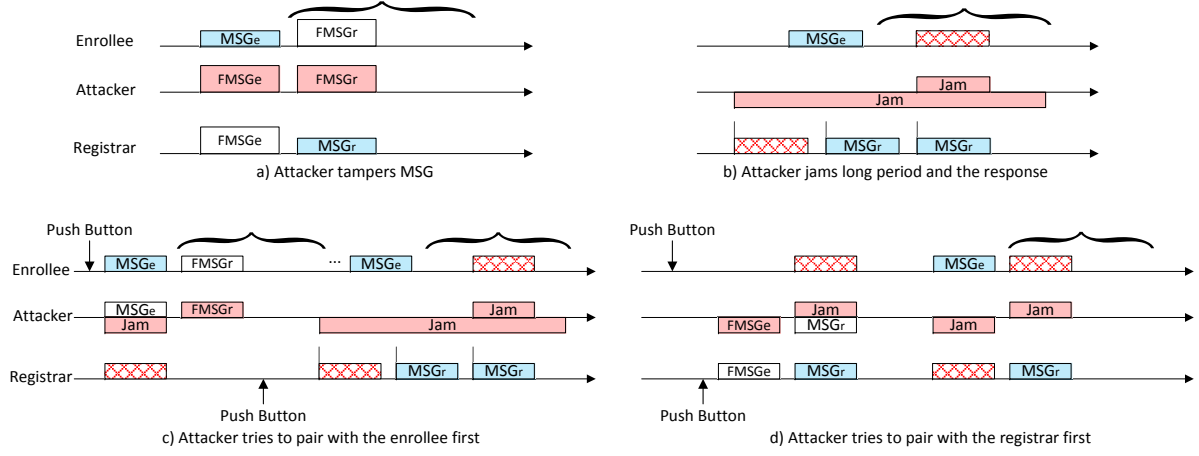


Figure 5.8: Attack scenarios. The buttons on the enrollee and the registrar were pushed in a) and b).

In this section, we will examine the behavior of our proposed protocol under different attack scenarios. Possible attack scenarios are shown in Fig. 5.8.

In Scenario a), the attacker tries to tamper the benign pairing messages by using a much stronger transmit power. However, the on-off sub-carriers are balanced and the attacker cannot transform an on sub-carrier to an off sub-carrier, the pairing parties (the enrollee and the registrar) are able to detect the mismatch between the hash computed from the payload and the hash conveyed by the on-off sub-carriers. As a result, the fake pairing messages  $FMSG_e$  and  $FMSG_r$  will be detected.

In Scenario b), in order to trick the registrar, the attacker will jam the registrar continuously before the enrollee's transmission, so that the response message  $MSG_r$  can fall out of the CTS reserved duration at the enrollee. To defend against this kind of attack, the CTS reserved duration  $T_s$  is set to be  $2T_m + 2T_{sifs}$ . Whenever the registrar detects a collision that is longer than a pairing message duration, it will wait for a SIFS duration and transmit its pairing message  $MSG_r$ . Therefore, at least one  $MSG_r$  will fall into the CTS reserved duration and the attacker has to jam it, which will cause a collision. Then the enrollee will be aware of the existence of the attacker.

Scenario c) shows that the attacker tries to pair with the enrollee before the button on the registrar is pushed, then jams the registrar to block  $MSG_e$ . As the enrollee runs the protocol 120

$sec + \#channel \times (T_m + T_s + T_w)$  before completing the pairing process, at least one  $MSG_e$  is transmitted after the registrar's button is pushed. Therefore, the enrollee can detect a collision in the CTS reserved duration and refuse to pair with the attacker.

In Scenario d), the attacker tries to pair with the registrar first, then jams  $MSG_e$ . The registrar will send its pairing message  $MSG_r$  upon detecting a collision that is longer than the pairing message, and the enrollee can detect the collision in CTS reserved time. Both of them will abort the pairing process due to collisions.

## 5.4 Analysis

In this section, we will first evaluate the efficiency of our proposed scheme by analyzing the channel occupation time, then give out formal proof to show the proposed scheme is secure to MITM attacks.

### 5.4.1 Channel Occupation Time

For illustration purpose, we assume that the size of the hash digest is 128 bits, which is balanced, padded to 156 bits, and divided into 3 groups (i.e.,  $156/52=3$ ). Because the CTS is much shorter than the message payload, without loss of generality, we omit it for brevity. Let  $l_m$  and  $r$  denote the benign pairing message payload length and the bit rate respectively. Further let  $T_h$  and  $T_m$  denote the channel occupation time for hash signals and the duration for a single pairing message respectively. We then have

$$T_m = \frac{l_m}{r} + T_h.$$

When using OFDM, the transmitter will send hash signal symbols of one group in parallel. Thus, the hash signals will occupy the channel for 3 symbol duration, each OFDM symbol in 802.11a is  $4 \mu s$ , then  $T_h = 12 \mu s$ . The pairing message length is set to be 3,000 bytes to allow the registrar to distinguish between the jammed pairing message and the ordinary collisions. Accordingly, the channel occupation time  $T_m$  is  $456.4 \mu s$  for a 54 Mbps 802.11a network. In other words, our pairing message reduces over 98% of the  $24,760 \mu s$  channel occupation time for one pairing message in TEP.

Note that at the enrollee, the channel will be held for the CTS reserved duration. To guarantee that the enrollee can receive the registrar's response, the CTS reserved duration  $T_s = 2 \cdot T_m + 2 \cdot T_{sifs}$ . Therefore, when we count the CTS reserved duration, the channel occupation time at the enrollee is about  $3 \cdot T_m + 2 \cdot T_{sifs}$ , which is still much shorter than  $24,760 \mu s$ .

## 5.4.2 Security Analysis

In this section, we formally prove that the proposed PBC scheme is secure to the MITM attacks.

**Proposition 5.4.1** *Given that the enrollee and the registrar are working properly and within the range of each other, any tampering or hiding attempts can be detected by the receiver (the enrollee or the registrar).*

*Proof.* As the attacker cannot estimate the channel between the enrollee and the registrar, we assume it cannot cancel out wireless signals between them as mentioned in Section 5.2.

First, as the on-off sub-carriers are balanced, to tamper the message successfully, the attacker must be able to alter an on sub-carriers to an off-one. This means the attacker must be able to cancel the signals on that sub-carriers, which contradicts with our assumption. Second, as the attacker cannot cancel the signals of the benign pairing message, it has to launch jamming to form a collision with the pairing message on the receiver side. As the benign pairing message is twice of a package duration, the receiver can distinguish it from other normal collisions, then the hiding attempt can be detected.

**Proposition 5.4.2** *Given that the enrollee and the registrar are working properly and within the range of each other, suppose that the enrollee and the registrar follow the protocol in Section 5.3.5, the proposed scheme is secure to MITM attacks.*

*Proof.* First, the registrar cannot be tricked to pair with the attacker falsely. As the timer in the protocol for the enrollee and the registrar are set to  $120 \text{ sec} + \# \text{ channels} \times (T_m + T_s + T_w)$  and the buttons of the enrollee and the registrar will be pushed within 120 seconds of each other, then at least one  $MSG_e$  is transmitted on the registrar's channel while the registrar is listening. According to Proposition 5.4.1, the attacker cannot tamper or hide the pairing message without being notice. In other words, if the attacker tries to tamper or hide  $MSG_e$ , in either case, the registrar can detect the collision (a tampered message is treated as a collision), and refuse to pair according to its protocol.

Second, suppose that the attacker convinces the enrollee to pair with it, which means the enrollee didn't receive any benign  $MSG_r$  or detect any collision during  $T_s$ . As the registrar will reply a  $MSG_r$  whenever it detects  $MSG_e$  or a collision, therefore the attacker either hid the  $MSG_e$  successfully so that the registrar didn't detect  $MSG_e$  or collisions, or hid the replied  $MSG_r$  so that the enrollee didn't receive  $MSG_r$  or collisions in  $T_s$ . Both of these two cases contradict with Proposition 5.4.1.

Therefore, the attacker cannot convince the enrollee or the registrar to pair with it, the proposed scheme is secure to MITM attacks.

## 5.5 Implementation and Evaluation

We have implemented a prototype based on GNURadio and USRP N210, and performed the real world experiments to evaluate our proposed techniques. The experimental results are summarized in this Section.

### 5.5.1 Prototype Setup

The prototype system contains two nodes to represent two pairing devices. Each node is implemented using a laptop running GNURadio code as the host machine and the USRP board with the XCVR 2450 daughter board that operates at the 2.4 GHz bands as its RF front end. We use the binary phase-shift keying (BPSK) for modulation and utilize both GNURadio and MATLAB for signal processing.

Our experiments use 128-bit hash digest. The 128-bit hash digest is encoded and padded to 156-bit, and then is divided into 3 groups, with each group containing 52 hash bits. To generate the hash signals for each group, the transmitter firstly uses BPSK to modulate the input hash bits to discrete base-band signals, and then uses the *ofdm\_carrier\_allocator* in GNURadio to map the signals to the 52 sub-carriers in an one-to-one way. Finally, the transmitter adopts IFFT to move the signals to different sub-carriers, as shown in Fig. 5.9.

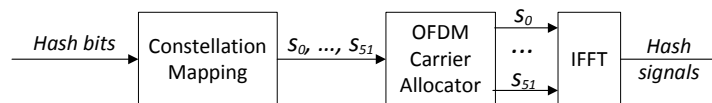


Figure 5.9: Hash signal generation.  $s_i$  is the discrete base-band signal.

On the other side, the receiver performs the FFT on the received hash signals, and then applies the energy detection on the on-off sub-carriers to de-map them to hash bits. In our implementation, we use 64-point IFFT and FFT for the signal processing, which is consistent with the 802.11 a/g OFDM implementation.

### 5.5.2 Accuracy Evaluation

We evaluate the accuracy of the proposed technique using the frequency domain sub-carriers to carry hash bits information.

In our experiment, the transmitter and the receiver are about three meters away from each other. The transmitter sends the modulated packet signals with the hash signals. On the other

side, the receiver firstly de-modulates the packet signals, and then uses the FFT to get the energy of different sub-carriers.

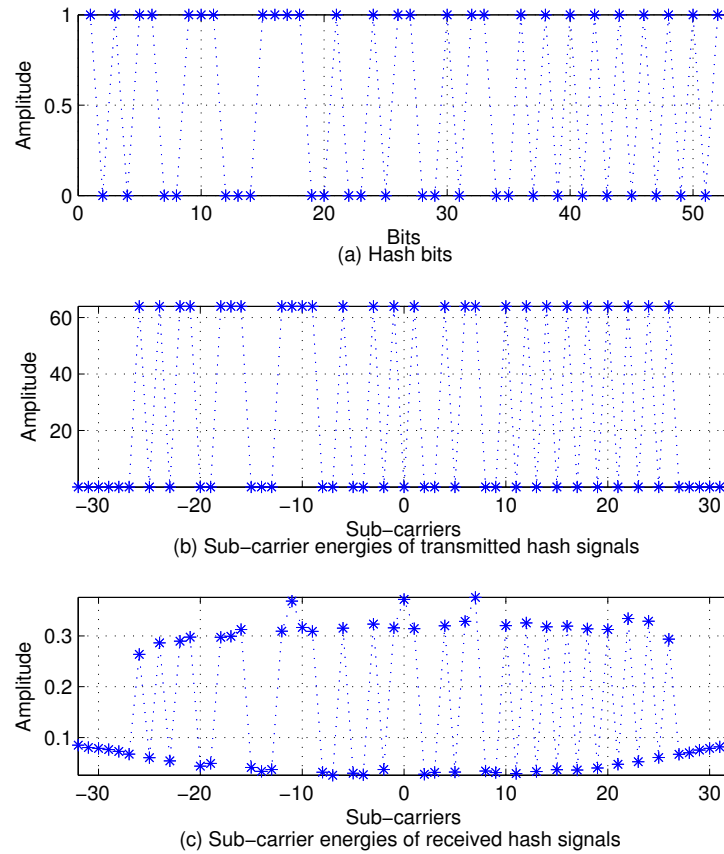


Figure 5.10: Hash bits and the sub-carrier energy.

Fig. 5.10 (a) shows the hash bits 1, 0, 1, 0, 1, 1, 0, 0, ... in one group. These 52 hash bits are mapped to the 52 sub-carriers in an one-to-one way. Fig. 5.10 (b) shows the energy of mapped sub-carriers on the transmitter side. We can see that the 52 hash bits are mapped to the central 52 sub-carriers (except for sub-carrier 0). The on sub-carrier (mapped to bit “1”) has high transmission energy while the off sub-carrier (mapped to bit “0”) has close-to-zero energy.

The transmitted hash signals go through the channel before being received by the receiver. The received signals are distorted by the channel and hardware effects such as the channel attenuation, phase shift, noise and multi-path fading. Fig. 5.10 (c) shows the FFT outcome of the received hash signals. Even though the received signals are impacted by the channel and

hardware effects, it is easy to see that the energy difference between the on sub-carrier and the off sub-carrier is obvious and the receiver can distinguish the on sub-carriers from the off sub-carriers readily.

We perform our experiments by letting the transmitter send 100 pairing packets. Each pairing packet contains 3,000 bytes payload and 3 groups of hash signals. For each group of the hash signals, the receiver conducts the 64-point FFT on the received hash signals to get the energy on the 52 sub-carriers. We compare the energy of the on sub-carriers and the off sub-carriers at the receiver. The result is shown in Fig. 5.11. We can see that the energy of the

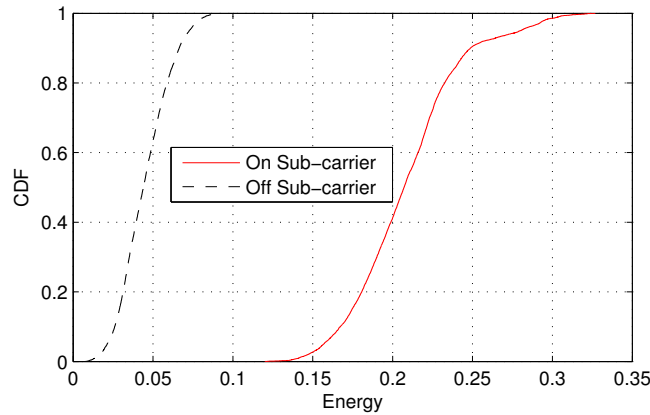


Figure 5.11: Energy CDF of on-off sub-carriers.

on sub-carrier is higher than the energy of the off sub-carriers and there is no overlap between the on sub-carriers' energy and the off sub-carriers' energy. Therefore, by enforcing a threshold, the receiver can distinguish the on sub-carrier from the off sub-carrier easily.

The receiver uses a threshold to de-map the energy of the sub-carrier to its carried hash bit. If the energy is larger than a threshold, the receiver deduces that this sub-carrier is an on sub-carrier and its carried hash bit is "1". Correspondingly, the sub-carrier with energy less than the threshold will be de-mapped to bit "0".

Note that at the receiver, the off sub-carriers may contain energy due to the noise and the energy leakage from adjacent sub-carriers. Therefore, when the threshold is too small, the off sub-carrier will be de-mapped to bit "1" mistakenly. On the contrary, when the threshold is too large, the on sub-carrier will be decoded as bit "0" falsely. Fig. 5.12 shows hash bit error rate w.r.t the energy threshold. We can observe that there is a range of threshold values which achieve close-to-zero hash bit error rate. In other words, the hash bits can be carried by using

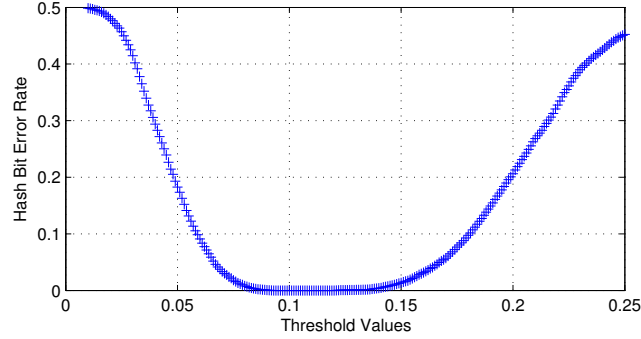


Figure 5.12: Hash bit error rates

the on-off sub-carriers in the frequency domain with close-to-zero bit errors. This means that the proposed technique of using sub-carriers to convey hash bit works accurately.

## 5.6 Related work

TEP [35] is the most closely related work to this work. As mentioned earlier, the long pairing message in TEP will interrupt the wireless connections and reduce the Wi-Fi throughput. Compared with TEP, our work uses the CTS and the on-off sub-carriers to secure the wireless in-band pairing process and reduce most of the channel occupation time.

This work is also related to the work on secure device pairing [19, 20, 70, 21]. On-off keying slots in the time domain were first used by Capkun et al. to protect the message integrity during the wireless transmission [19, 20]. The Good Neighbor scheme was later proposed in [21] to securely pair two nearby devices by exploiting the multiple antennas capability. There are also studies of using out-of-band channels to pair devices securely. The out-of-band channels can be created by leveraging a camera [56, 72], a microphone [60, 66], an accelerometer [39, 55], and an infra-red channel [16]. Our work removes the need of using out-of-band channels, which makes it more suitable for PBC devices that have very simple user interfaces.

Recently, there have been multiple research work which explores the new usage of OFDM sub-carriers. For example, in order to improve the 802.11 network throughput, Back2F [73] proposed to use the sub-carrier index to replace the real backoff time in a 802.11 wireless network. Dutta et al. proposed to use OFDM sub-carriers to achieve simultaneous transmissions [27]. MCBC [69] proposed to use short and un-modulated burst of energy on the sub-carriers for the contention of medium access control. FICA [82] proposed to divide the channel into sub-channels and uses the frequency domain back off to coordinate sub-channel access. All these works are orthogonal to our work.

## Chapter 6

# Conclusion and Future Work

### 6.1 Conclusion

This dissertation contains four works towards leveraging physical layer signals manipulation to achieve wireless physical layer security. In the first work, we presented MCR decoding, a technique aiming at providing the anti-jamming wireless communication capability for multi-antenna wireless devices. To perform MCR decoding, the receiver monitors the change of MCR values to detect the jammed ongoing transmission, then applies the jammer's MCR value to remove the jamming signals. We have implemented and evaluated MCR decoding on GNURadio and USRP. Our experimental results showed that MCR decoding can detect the desired transmission reliably under jamming attacks and remove more than 99.86% of the jamming signal power in the real world environment.

In the second work, we proposed ally friendly jamming, a mechanism that jams unauthorized wireless communication and maintains legitimate communication at the same time. Ally friendly jamming is achieved by properly controlling the ally jamming signals using secret keys shared among authorized devices and ally jammers. We have analyzed the properties of ally friendly jamming, implemented a prototype system, and performed a series of experimental evaluations. Our results demonstrated that the proposed techniques can effectively disable unauthorized wireless communication and at the same time allow wireless communication between authorized devices.

In the third work, we proposed fast friendly jamming, a novel design to allow the friendly jammer to distinguish the authorized and unauthorized wireless transmissions through verifying auth-preamble signals on the physical layer. We have implemented a prototype of the proposed techniques and performed real-world experiments to evaluate the performance. The experimental results showed that the proposed techniques reduce the reaction delay of the friendly jammer by 81.9% – 85.7% as compared to the traditional demodulation methods, and enable

the accurate distinction between authorized and unauthorized transmissions.

In the fourth work, we came up with a novel wireless in-band pairing design, in which one pairing node sends back its pairing messages upon detecting potential jammed pairing messages and the other pairing node utilizes the CTS reserved duration to detect possible attackers in the network. We also proposed to use the on-off sub-carriers in the frequency domain to convey the hash digest of the pairing message from the sender to the receiver, to protect the integrity of a benign pairing message.

Our design reduces most of the channel occupation time by eliminating the long synchronization packet and the on-off slots in the time domain of TEP, and hence decreases the wireless medium contention impacts on the local Wi-Fi networks. We have implemented a prototype system of the proposed techniques based on GNURadio and performed experiments to validate the proposed techniques. The experimental results showed that the proposed technique of using sub-carriers in the frequency domain to convey hash bits information works accurately in the real-world environment.

## 6.2 Future Work

Based on discussions in the previous sections, we propose three future research directions:

- **Target ally friendly jamming on a specific kind of wireless signal.** The ally friendly jamming technique we proposed is a generic framework for a variety of wireless applications. The ally jammer is designed to block all kinds of wireless signals. However, different wireless signals have different characteristics, which can be exploited by the ally jammer to achieve better jamming performance. How to use these characteristics to achieve maximum jamming effect and maintain the ally's wireless communication at the same time is a challenging task. New techniques about ally jamming signal generation, device synchronization and jamming signal removal are needed.

One specific work can be extending ally friendly jamming to GPS signals, to achieve ally friendly GPS. The GPS signals utilize direct sequence spread spectrum (DSSS) modulation, which is controlled by a set of spreading codes. The ally jammer can use the same set of spreading codes to generate its jamming signals to maximize the jamming performance. This work is highly desirable as it provides the flexibility to turn off the GPS service in certain regions targeting certain devices.

- **Extend ally friendly jamming to MIMO devices.** Now, with MIMO devices becoming more and more popular, ally friendly jamming should also be extended to MIMO devices, including authorized/unauthorized MIMO devices and MIMO ally jammers. One possible way of extending the current approach to the MIMO ally jammer case is: using

a different key to generate jamming signals on each of the transmit paths of a MIMO ally jammer, and let the authorized receiver treat the MIMO ally jammer as multiple ally jammers. More studies are required for the cases of authorized/unauthorized MIMO devices.

- **Generalize fast friendly jamming to the multi-tap channel.** Currently the fast friendly jamming removes channel effects on the received signals by utilizing the linear channel coefficient. However, when the channel is multi-taped, channel coefficients are not linear. Therefore, new techniques need to be explored in order to extend fast friendly jamming to multi-tap channel scenarios.

## REFERENCES

- [1] Dropcam-wi-fi video monitoring. <https://www.dropcam.com>.
- [2] Gnu radio - the gnu software radio. <http://gnuradio.org/redmine/projects/gnuradio/wiki>.
- [3] Home monitor-wi-fi wireless video. <http://www.homemonitor.me/>.
- [4] Improvised explosive device - wikipedia. [http://en.wikipedia.org/wiki/Improvised\\_explosive\\_device](http://en.wikipedia.org/wiki/Improvised_explosive_device).
- [5] Medical device networking. <http://www.silexamerica.com/wireless-expertise/medical-devices/>.
- [6] Orthogonal frequency-division multiplexing. [http://en.wikipedia.org/wiki/Orthogonal\\_frequency-division\\_multiplexing](http://en.wikipedia.org/wiki/Orthogonal_frequency-division_multiplexing).
- [7] Ppm. [http://en.wikipedia.org/wiki/Parts\\_per\\_million](http://en.wikipedia.org/wiki/Parts_per_million).
- [8] Q-function. <http://en.wikipedia.org/wiki/Q-function>.
- [9] Usrc n210 datasheet. [https://www.ettus.com/content/files/07495\\_Ettus\\_N200-210\\_DS\\_Flyer\\_HR\\_1.pdf](https://www.ettus.com/content/files/07495_Ettus_N200-210_DS_Flyer_HR_1.pdf).
- [10] The usrp product family products and daughter boards - ettus research LLC. <http://www.ettus.com/products>.
- [11] Wi-fi protected setup. [http://en.wikipedia.org/wiki/Wi-Fi\\_Protected\\_Setup](http://en.wikipedia.org/wiki/Wi-Fi_Protected_Setup).
- [12] Wi-Fi alliance. Wi-fi protected setup specification, version 1.0h. 2006.
- [13] Wi-Fi alliance. Wi-fi in healthcare: Security solutions for hospital wi-fi networks. 2012.
- [14] E. Aryafar, N. Anand, T. Salonidis, and E. W. Knightly. Design and experimental evaluation of multi-user beamforming in wireless LANs. In *MobiCom*, 2010.

- [15] L. C. Baird, W. L. Bahn, M. D. Collins, M. C. Carlisle, and S. C. Butler. Keyless jam resistance. In *IEEE Information Assurance and Security Workshop*, 2007.
- [16] D. Balfanz, G. Durfee, D.K. Smetters, and R. Grinter. In search of usable security: Five lessons from the field. *IEEE Journal on Security and Privacy*, 2(5):19–24, 2004.
- [17] Dinesh Bharadia, Emily McMilin, and Sachin Katti. Full duplex radios. In *Proceedings of ACM SIGCOMM*, 2013.
- [18] L. Buttyan, D. Gessner, A. Hessler, and P. Langendoerfer. Application of wireless sensor networks in critical infrastructure protection: challenges and design options. *IEEE Wireless Communications*, 17(5), 2010.
- [19] M. Cagalj, S. Capkun, and J-P Hubaux. Key agreement in peer-to-peer wireless networks. In *Proceedings of IEEE (Special Issue on Cryptography and Security)*, 2006.
- [20] Mario Cagalj, Srdjan Capkun, Ramkamur Rengaswamy, Ilias Tsigkogiannis, Mani Srivastava, and J-P Hubaux. Integrity (i) codes: Message integrity protection and authentication over insecure channels. In *Proceedings of IEEE Symposium on Security and Privacy*, 2006.
- [21] Liang Cai, Kai Zeng, Hao Chen, and Prasant Mohapatra. Good neighbor: Secure pairing of nearby wireless devices by multiple antennas. In *Proceedings of Network and Distributed Systems Security Symposium*, 2011.
- [22] P. Castoldi. *Multiuser detection in CDMA mobile terminals*. Artech house Publishers, 2002.
- [23] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *Proceedings of the 24th IEEE Symposium on Security and Privacy*, 2003.
- [24] B. DeBruhl and P. Tague. Digital filter design for jamming mitigation in 802.15.4 communication. In *ICCCN*, 2011.

- [25] B. DeBruhl and P. Tague. Mitigation of periodic jamming in a spread spectrum system by adaptive filter selection. In *PECCS*, 2012.
- [26] Whitfield Diffie and Martin E Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.
- [27] Aveek Dutta, Dola Saha, Dirk Grunwald, and Douglas Sicker. Smack: a smart acknowledgment scheme for broadcast messages in wireless networks. In *Proceedings of ACM SIGCOMM Computer Communication Review*, 2009.
- [28] M. Erol-Kantarci and H.T. Mouftah. Wireless multimedia sensor and actor networks for the next generation power grid. *Ad Hoc Networks*, 2011.
- [29] D. Gesbert, M. Shafi, D. Shiu, P.J. Smith, and A. Naguib. From theory to practice: an overview of MIMO space-time coded wireless systems. *IEEE JSAC*, 2003.
- [30] A. Goldsmith. *Wireless communications*. Cambridge University Press, 2005.
- [31] S. Gollakota, F. Adib, D. Katabi, and S. Seshan. Clearing the RF smog: Making 802.11 robust to cross-technology interference. In *SIGCOMM*, 2011.
- [32] S. Gollakota, H. Hassanieh, B. Ransford, D. Katabi, and K. Fu. They can hear your heartbeats: non-invasive security for implantable medical devices. In *SIGCOMM*, 2011.
- [33] S. Gollakota and D. Katabi. Zigzag decoding: Combating hidden terminals in wireless networks. In *SIGCOMM*, 2008.
- [34] S. Gollakota, S.D. Perli, and D. Katabi. Interference alignment and cancellation. In *SIGCOMM*, 2009.
- [35] Shyamnath Gollakota, Nabeel Ahmed, Nickolai Zeldovich, and Dina Katabi. Secure in-band wireless pairing. In *Proceedings of USENIX Security Symposium*, 2011.
- [36] D. Halperin, T. Anderson, and D. Wetherall. Taking the sting out of carrier sense: interference cancellation for wireless LANs. In *MobiCom*, 2008.

- [37] S.S. Haykin. *Digital communications*, volume 5. Wiley, 1988.
- [38] X. He, H. Dai, and P. Ning. Dynamic adaptive anti-jamming via controlled mobility. In *Proceedings of 2013 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9. IEEE, 2013.
- [39] Lars Erik Holmquist, Friedemann Mattern, Bernt Schiele, Petteri Alahuhta, Michael Beigl, and Hans-W Gellersen. Smart-its friends: A technique for users to easily establish connections between smart artefacts. In *Proceedings of ACM Conference on Pervasive and Ubiquitous Computing*, 2001.
- [40] IEEE. IEEE 802.11: Wireless LANs, 2007.
- [41] Y. Kim, A. Perrig, and G. Tsudik. Simple and fault-tolerant key agreement for dynamic collaborative groups. In *Proceedings of the ACM Conference on Computer and Communications Security*, 2000.
- [42] Cynthia Kuo, Jesse Walker, and Adrian Perrig. Low-cost manufacturing, usability, and security: an analysis of bluetooth simple pairing and wi-fi protected setup. In *Financial Cryptography and Data Security*. 2007.
- [43] James F. Kurose and Keith W. Ross. *Computer networking: a top-down approach featuring the Internet*. Pearson, 2012.
- [44] B. Latre, B. Braem, I. Moerman, C. Blondia, and P. Demeester. A survey on wireless body area networks. *Wireless Networks*, 17(1), 2011.
- [45] R.C.T. Lee, M.C. Chiu, and J.S. Lin. *Communications engineering: Essentials for computer scientists and electrical engineers*. John Wiley & Sons, 2007.
- [46] K.C. Lin, S. Gollakota, and D. Katabi. Random access heterogeneous MIMO networks. In *SIGCOMM*, 2011.

- [47] A. Liu, P. Ning, H. Dai, and Y. Liu. USD-FH: Jamming-resistant wireless communication using frequency hopping with uncoordinated seed disclosure. In *MASS*, 2010.
- [48] A. Liu, P. Ning, H. Dai, Y. Liu, and C. Wang. Defending DSSS-based broadcast communication against insider jammers via delayed seed-disclosure. In *ACSAC*, 2010.
- [49] D. Liu, P. Ning, and R. Li. Establishing pairwise keys in distributed sensor networks. *ACM Transactions on Information and System Security*, 2005.
- [50] D. Liu, P. Ning, and K. Sun. Efficient self-healing group key distribution with revocation capability. In *Proceedings of the ACM Conference on Computer and Communications Security(CCS)*, 2003.
- [51] Y. Liu, P. Ning, H. Dai, and A. Liu. Randomized differential DSSS: Jamming-resistant wireless broadcast communication. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, 2010.
- [52] Yao Liu, Peng Ning, and Huaiyu Dai. Authenticating primary users' signals in cognitive radio networks via integrated cryptographic and wireless link signatures. In *Proceedings of IEEE Symposium on Security and Privacy*, 2010.
- [53] R.G. Lyons. *Understanding digital signal processing*. Prentice Hall, 2011.
- [54] I. Martinovic, P. Pichota, and J. B. Schmitt. Jamming for good: a fresh approach to authentic communication in WSNs. In *WiSec*, 2009.
- [55] Rene Mayrhofer and Hans Gellersen. Shake well before use: Intuitive and secure pairing of mobile devices. *IEEE Transactions on Mobile Computing*, 2009.
- [56] Jonathan M McCune, Adrian Perrig, and Michael K Reiter. Seeing-is-believing: Using camera phones for human-verifiable authentication. In *Proceedings of IEEE Symposium on Security and privacy*, 2005.

- [57] H. Meyr, M. Moeneclaey, and S.A. Fechtel. *Digital communication receivers : synchronization, channel estimation, and signal processing*. John Wiley & Sons, 1998.
- [58] Muhammad Naveed, Xiaoyong Zhou, Soteris Demetriou, XiaoFeng Wang, and Carl A Gunter. Inside job: Understanding and mitigating the threat of external device mis-bonding on android. In *Proceedings of the Network and Distributed Systems Security Symposium*, 2014.
- [59] K. Pahlavan and P. Krishnamurthy. *Principles of wireless networks*. Prentice Hall, 2001.
- [60] Chunyi Peng, Guobin Shen, Yongguang Zhang, and Songwu Lu. Point&connect: intention-based device pairing for mobile phone users. In *Proceedings of Conference on Mobile Systems, Applications, and Services*, 2009.
- [61] A. Perrig, D. Song, and J.D. Tygar. ELK, a new protocol for efficient large-group key distribution. In *Proceedings of the 22nd IEEE Symposium on Security and Privacy*, 2001.
- [62] R. Poisel. *Modern Communications Jamming Principles and Techniques*. Artech House Publishers, 2006.
- [63] C. Pöpper, M. Strasser, and S. Čapkun. Jamming-resistant broadcast communication without shared keys. In *USENIX Security Symposium*, 2009.
- [64] C. Pöpper, M. Strasser, and S. Čapkun. Anti-jamming broadcast communication using uncoordinated spread spectrum techniques. *IEEE JSAC*, 2010.
- [65] J.G. Proakis and M. Salehi. *Digital communications*. McGraw-hill, 2008.
- [66] Kasper Bonne Rasmussen, Claude Castelluccia, Thomas S Heydt-Benjamin, and Srdjan Capkun. Proximity-based access control for implantable medical devices. In *Proceedings of ACM Conference on Computer and Communications Security*, 2009.
- [67] M. Rieback, B. Crispo, and A. Tanenbaum. RFID guardian: A battery-powered mobile device for rfid privacy management. In *Information Security and Privacy*. Springer, 2005.

- [68] M. Rieback, B. Crispo, and A. Tanenbaum. Keep on blockin in the free world: Personal access control for low-cost rfid tags. In *Security Protocols*, pages 51–59. Springer, 2007.
- [69] Bogdan Roman, Frank Stajano, Ian Wassell, and David Cottingham. Multi-carrier burst contention (mcbc): Scalable medium access control for wireless networks. In *Proceedings of the Wireless Communications and Networking Conference*, 2008.
- [70] Volker Roth, Wolfgang Polak, Eleanor Rieffel, and Thea Turner. Simple and effective defense against evil twin access points. In *Proceedings of ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2008.
- [71] S. Sankararaman, K. Abu-Affash, A. Efrat, S. D. Eriksson-Bique, V. Polishchuk, S. Ramasubramanian, and M. Segal. Optimization schemes for protective jamming. In *MobiHoc*, 2012.
- [72] Nitesh Saxena, J-E Ekberg, Kari Kostianen, and N Asokan. Secure device pairing based on a visual channel. In *Proceedings of IEEE Symposium on Security and Privacy*, 2006.
- [73] Souvik Sen, Romit Roy Choudhury, and Srihari Nelakuditi. No time to countdown: Migrating backoff to the frequency domain. In *Proceedings of Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2011.
- [74] Wenbo Shen, Yao Liu, Xiaofan He, Huaiyu Dai, and Peng Ning. No time to demodulate - fast physical layer verification of friendly jamming. In *Military Communications Conference (MILCOM)*. IEEE, 2015.
- [75] Wenbo Shen, Peng Ning, Xiaofan He, and Huaiyu Dai. Ally friendly jamming: How to jam your enemy and maintain your own wireless connectivity at the same time. In *IEEE Symposium on Security and Privacy*, 2013.
- [76] Wenbo Shen, Peng Ning, Xiaofan He, Huaiyu Dai, and Yao Liu. MCR decoding: A MIMO approach for defending against wireless jamming attacks. In *Proceedings of IEEE CNS workshop on Physical-layer Methods for Wireless Security*, 2014.

- [77] D. Slater, P. Tague, R. Poovendran, and M. Li. A game-theoretic framework for jamming attacks and mitigation in commercial aircraft wireless networks. In *AIAA Infotech@Aerospace Conference*, 2009.
- [78] D. Slater, P. Tague, R. Poovendran, and B. Matt. A coding-theoretic approach for efficient message verification over insecure channels. In *WiSec*, 2009.
- [79] M. Strasser, C. Pöper, S. Čapkun, and M. Čagalj. Jamming-resistant key establishment using uncoordinated frequency hopping. In *IEEE Symposium on Security and Privacy*, 2008.
- [80] M. Strasser, C. Pöpper, and S. Čapkun. Efficient uncoordinated FHSS anti-jamming communication. In *MobiHoc*, 2009.
- [81] K. Tan, H. Liu, J. Fang, W. Wang, J. S. Zhang, M. Chen, and G. M. Voelker. SAM: Enabling practical spatical multiple access in wireless LAN. In *MobiCom*, 2009.
- [82] Kun Tan, Ji Fang, Yuanyang Zhang, Shouyuan Chen, Lixin Shi, Jiansong Zhang, and Yongguang Zhang. Fine-grained channel access in wireless LAN. *ACM SIGCOMM Computer Communication Review*, 2011.
- [83] D. Thunte and M. Acharya. Intelligent jamming in wireless networks with applications to 802.11 b and other networks. In *Proc. of MILCOM*, volume 6, 2006.
- [84] N. O. Tippenhauer, L. Malisa, A. Ranganathan, and S. Capkun. On limitations of friendly jamming for confidentiality. In *IEEE Symposium on Security and Privacy*, 2013.
- [85] M. Wilhelm, I. Martinovic, J. B Schmitt, and V. Lenders. Short paper: reactive jamming in wireless networks: how realistic is the threat? In *Proceedings of the fourth ACM conference on Wireless network security*, pages 47–52. ACM, 2011.
- [86] M. Wilhelm, I. Martinovic, J. B Schmitt, and V. Lenders. Wifire: A firewall for wireless networks. In *ACM SIGCOMM Computer Communication Review*, volume 41, pages 456–457. ACM, 2011.

- [87] D. Willkomm, J. Gross, and A. Wolisz. Reliable link maintenance in cognitive radio systems. In *DySPAN*, 2005.
- [88] C. K. Wong, M. G. Gouda, and S. S. Lam. Secure group communications using key graphs. In *SIGCOMM*, 1998.
- [89] L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe. Fingerprints in the ether: Using the physical layer for wireless authentication. In *Proceedings of IEEE International Conference on Communications (ICC)*, 2007.
- [90] L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe. Using the physical layer for wireless authentication in time-variant channels. *IEEE Transactions on Wireless Communications*, 2008.
- [91] F. Xu, Z. Qin, C. C. Tan, B. Wang, and Q. Li. IMDGuard: Securing implantable medical devices with the external wearable guardian. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, 2011.
- [92] W. Xu, W. Trappe, Y. Zhang, and T. Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *MobiHoc*, 2005.
- [93] S. Yoon, B. Jung, K Lee, and I. Rhee. Adopt: Practical add-on MIMO receiver for concurrent transmissions. Technical report, NCSU, 2012.
- [94] P. L Yu, J. S Baras, and B. M Sadler. Physical-layer authentication. *IEEE Transactions on Information Forensics and Security*, 2008.
- [95] Q. Zhu, H. Li, Z. Han, and T. Basar. A stochastic game model for jamming in multi-channel cognitive radio systems. In *ICC*, 2010.