AUTHENTICATION AND MESSAGE INTEGRITY VERIFICATION WITHOUT SECRETS

by

Nirnimesh Ghose

A Dissertation Submitted to the Faculty of the

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

In Partial Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

2019

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation prepared by *Nirnimesh Ghose*, titled *Authentication and Message Integrity Verification without Secrets* and recommended that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Dr. Loukas Lazos — Date: 04/12/2019

Dr. Marwan Krunz — Date: 04/12/2019

Dr. Ming Li — Date: 04/12/2019

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommended that it be accepted as fulfilling the dissertation requirement.

Dr. Loukas Lazos — Date: 04/12/2019
Associate Professor
Department of Electrical and Computer Engineering

ACKNOWLEDGEMENTS

First of all, I would like to express my deepest gratitude to my advisor, Professor Loukas Lazos, for all the help, advice, patience, support, and encouragement he gave to me during the past six years. Because of him, my graduate experience has been one I will cherish forever. I am fortunate to have him as my advisor. He enlightened me through his professional knowledge about where to explore, how to think/research, and what is necessary to get there. This dissertation would have been impossible without his guidance.

I am grateful to Professor Ming Li for his guidance. He has been co-advising me and providing insightful inputs in several works of this dissertation. I have also enjoyed many long hours of discussions with him on various topics.

I would like to thank Professor Marwan Krunz and Professor Ming Li for serving on my Ph.D. committee and for providing all the insightful comments, valuable suggestions, and discussions that make this dissertation better. I would also like to thank my minor advisor Professor Wei Lin Hua for helping me complete my minor courses. Your time and effort are greatly appreciated.

I would like to thank my labmates and colleagues for their help and friendship, including Alejandro Proano, Yan Zhang, Swetha Shivaramaiah, Nicholas Fragiskatos, Jose Carlos Acedo, Bocan Hu, Kai Chen, Li Li, Islam Samy, and Xiao Han. I would also thank Tami Whelan for handling all the paperwork and giving various forms of support during my graduate study.

To my Maa, The Late Dr. Ranjana Ghose, my wife Jayna Rose Ghose, my sisters Dr. Tandra Ghose and Dr. Toolika Ghose, and my in-Laws James L. Sullivan, Diane M. Sullivan, Dr. Rupak Majumdar and Dr. Nikhil Satyala: Thank you for supporting and encouraging me throughout my Ph.D. study. Your love and care give me the strength to overcome difficulties throughout this endeavor. You have always been my source of inspiration, confidence, and success.

Finally, I would like to acknowledge the US National Science Foundation (NSF) and the US Army Research Office (ARO) for the financial support provided to conduct the research needed for this dissertation. This research was supported in part by the CNS-1409172, CNS-1731164, CNS-1347075, and ARO grant W911NF-13-1-0302. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF and the ARO.

DEDICATION

To my Maa, The Late Dr. Ranjana Ghose and my wife Jayna Rose Ghose.

TABLE OF CONTENTS

TABLE OF CONTENTS – *Continued*

TABLE OF CONTENTS – *Continued*

TABLE OF CONTENTS – *Continued*

TABLE OF CONTENTS – *Continued*

LIST OF FIGURES

LIST OF FIGURES – *Continued*

LIST OF FIGURES – *Continued*

LIST OF FIGURES – *Continued*

LIST OF FIGURES – *Continued*

LIST OF FIGURES – *Continued*

LIST OF FIGURES – *Continued*

LIST OF FIGURES – *Continued*

LIST OF TABLES

ABSTRACT

Embedding network capabilities in a plethora of new devices and infrastructures–the Internet-of-Things, vehicular and aviation networks, the critical national infrastructure, industrial plants–are dramatically transforming the modern way of living. The rapid deployment pace of these emerging applications has brought unprecedented security challenges related to data confidentiality, user privacy, and critical infrastructure availability. A significant portion of these threats is attributed to the broadcast nature of the wireless medium, which exposes systems to easy-to-launch passive and active attacks. The slow security standards rollout combined with the ever-shrinking time-to-market, the device heterogeneity and the lack of user-friendly input interfaces (screen, keyboard, etc.) only exacerbate the security challenges.

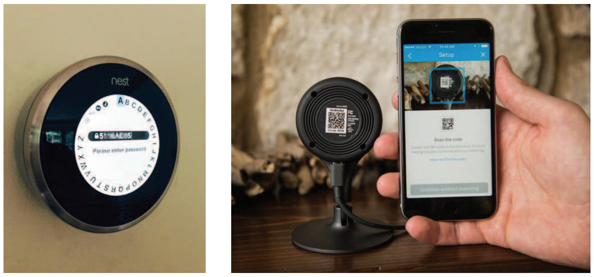
In this dissertation, we address the fundamental problem of trust establishment in the context of emerging network applications. We present techniques integrating physical layer properties with cryptographic primitives to guarantee message integrity and bootstrap initial trust without relying on any prior secrets. We present the "helper" security paradigm in which security is outsourced to one or more dedicated devices to allow for the scalable pairing of off-the-shelf heterogeneous devices. In addition, we present our work on message integrity verification of navigation information for aircrafts (speed, location, and heading) by exploiting the Doppler spread of the wireless channel. Finally, we develop a secure and fast voting technique for distributed networks which allows fast coordination of a group of devices without the overhead of messaging.

CHAPTER 1

INTRODUCTION

Embedding network capabilities in most contemporary systems has fundamentally changed the course of our everyday lives. A plethora of new services and infrastructures–the smart grid, transportation, and aviation networks, the cloud, the critical national infrastructure, the Internet-of-Things, and how we interact with our physical world–have dramatically transformed the modern way of living [1–6]. With the improvement in utility, our dependence on this rapidly expanding network ecosystem has brought unprecedented security challenges related to user privacy, data confidentiality, and critical infrastructure availability. As most of our communications occur in wireless platforms, new serious threats have emerged that cause widespread disruption and substantial monetary losses [7–10]. A significant portion of these threats is attributed to the broadcast nature of the wireless medium, which exposes systems to easy-to-launch passive and active attacks. Using commodity radio hardware, unauthorized parties can easily eavesdrop on or modify over-the-air transmissions [11–14]. In addition, the rapid commercialization of new devices and systems does not allow for careful consideration of the possible threats, with security only being an afterthought. Security challenges are only exacerbated by the lag in standardization efforts, the plethora of vendors flooding the market with poorly engineered devices, and the complex nature of today's security solutions (e.g., password etiquette and two-factor authentication) [15–17]. In such a heterogeneous environment, classical cryptographic solutions alone cannot meet the scalability, interoperability, usability, and most importantly, security requirements.

My dissertation is focused on answering fundamental security problems that lie in the intersection between privacy, security, usability, and efficiency. These include the basic problem of establishing trust between two or more legitimate parties in



(a) (b)

Figure 1.1: (a) Entering password to pair a smart thermostat with a network, and (b) scanning QR code with embedded default password to pair a smart camera with a network.

heterogeneous networks; how to seamlessly integrate security services without requiring user involvement; how to make security scalable and interoperable; how to retrofit security in already deployed systems; and how to automatically recover from unavoidable security breaches.

Classic techniques for secure pairing either involve the manual input shown in Figure 1.1(a) of the hub's secret to the device or the preloading of a unique secret. This secret is loaded to the hub via an out-of-band (OOB) channel, e.g., the user scanning QR code with embedded password shown in Figure 1.1(b), a public key infrastructure [18]. Nevertheless, traditional solutions pose significant usability, scalability, and interoperability hurdles. Alternatively, several device pairing protocols have been proposed for without pre-shared secrets [19–30]. Most such protocols require an auxiliary secure out-of-band (OOB) channel, an audio or visual channel, for example, that is observable by a user to aid the authentication of messages transmitted over the public wireless channel. However, such OOB channels introduce practical interoperability issues due to the heterogeneity of the devices and are not user-friendly. Recently, in-band pairing protocols [31–35] have been proposed as an alternative to OOB pairing. The former protocols only require that devices possess a common wireless interface to communicate. Since the wireless channel is known to be insecure in general, the security of these protocols relies on the assumption that wireless signal cancellation is infeasible, so that message integrity and

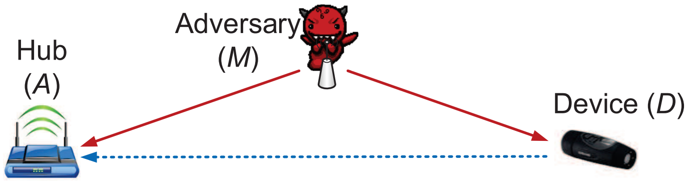


Figure 1.2: Trust establishment between a device with a hub in presence on an adversary.

authentication properties can be derived by encoding the messages in a special way. However, as demonstrated by Popper *et al.* [14], this assumption may not hold in many cases. Thus, it remains an open problem as to whether secure in-band device pairing protocols can still be designed under a strong Dolev-Yao attacker which can annihilate wireless signals.

To address these challenges, we proposed original techniques for pairing devices that lack preloaded security credentials by exploiting hard-to-forge physical limitations in signal propagation laws. Our techniques can withstand both passive and active attacks and are even suitable for devices without advanced interfaces such as screens and keyboards. We also proposed methods for retrofitting integrity verification in aviation navigation broadcasts without relying on cryptographic primitives. Such broadcasts are currently transmitted in the clear and are easy to forge. We have also integrated security mechanisms at the lower layers of the network stack to enable the fast and secure collaboration of groups of devices such as network-enabled Unmanned Aerial Vehicles (UAVs), distributed spectrum sensing, and wireless sensor networks.

1.1 Main Contributions

1.1.1 Trust Establishment

From logging in to our personal computers, to gaining access to our mobile devices, to accessing bank accounts, and more recently accessing data collected by wearables, thermostats, remotely opening garage doors, we are constantly asked to prove our identity. This is typically achieved by entering a password or a pin, verifying a fingerprint, or performing face recognition [36]. Such methods of proving one's identity, also known as user authentication, rely on advanced interfaces such as keyboards, cameras, fingerprint readers, and other biometric sensors. With the advent of Internet-of-Things and the miniaturization of networked devices, such interfaces may no longer be available. In general, the network enabled device and the hub has to prove mutual authenticity in the presence of an adversary as shown in Figure 1.2. Conventional authentication solutions include the use of default passwords [37], the preloading of common secrets, or the establishment of a public key infrastructure [18, 38]. However, such solutions pose serious key management, scalability, and interoperability challenges. Often, manufacturers opt to preload devices with default keys that are easily leaked [8]. Moreover, without keyboards and screens default passwords are hard to change. To address these challenges, recent works have proposed secure device pairing methods that do not rely on pre-shared secrets [23, 32, 34, 35, 39–41]. Most rely on out-of-band (OOB) human verification to provide authentication and verify the protocol success. Human-dependent solutions scale poorly with the number of devices. Some in-band solutions have also appeared, but they almost unanimously derive security from the *infeasibility of advanced wireless signal manipulations.* However, for an advanced adversary capable of manipulating wireless transmissions [14], the current state-of-the-art solutions are vulnerable to active attacks that breach the authentication process. We addressed this problem in three different scenarios:

Trust Establishment Under Active Signal Manipulations

We proposed a PHY layer-based message integrity assurance mechanism by detecting active signal manipulations, including hard-to-detect signal cancellation attacks. To thwart an adversary that is capable of perfect signal cancellation, we exploited a helper which is co-present with the legitimate device by letting the helper insert authentication signals onto the channel at random times. The security of our primitive rested on the infeasibility of differentiating the helper's and device's activities in real time. We used this message integrity verification primitive in conjunction with commonly-used key establishment protocols (e.g., the Diffie-Hellman key exchange protocol) to implement a complete trust establishment protocol.

Modulation-Agnostic Secure Trust Establishment

The majority of existing integrity verification methods that do not rely on pre-shared secrets [31–35] utilize a specific type of digital modulation. Although ON-OFF keying can be implemented from common modulation modes (BPSK, QPSK, ASK, etc.), it is not necessarily natively supported by legacy and new devices. Thus, the adoption of those methods may require firmware/hardware modifications. To be compatible with Commercial-Off-The-Shelf (COTS) devices, we proposed a novel message integrity and authentication primitive that does not rely on any special modulation. The core principle of our approach is to create a "received signal strength (RSS) authenticator" by analyzing the RSS fluctuation patterns, measured simultaneously at the helper placed close to the device and the hub. We concentrated our efforts on RSS because it is readily available on any wireless device and can be acquired in-band. Whereas some previous methods have tried to exploit RSS for device pairing, security was shown only if some specific channel conditions were met (e.g., rich scattering environments). The security of our method relies on hard-to-break physical signal propagation laws such as requiring extremely high transmission power, and the inability of an adversary to predict the helper's motion in real time.

Group Device Pairing Resistant to Active Signal Manipulations

In several scenarios, a group of devices is initialized within a short time. For such scenarios, executing pairwise pairing poses scalability challenges. In addition, the success probability of an adversary to pair with the legitimate setting increases exponentially with the number of devices in the group. We proposed a novel primitive to verify the integrity of messages transmitted among a group of devices, without preloaded secrets. Our method hardened active attacks by exploiting the simultaneous integrity verification of the protocol transcripts from multiple devices including the helper. During the pairing, each device broadcasts their primitive which is recorded by all other devices in the group (a helper device is included if the number of devices is less than the required number) to compile a transcript. Further to verify the integrity of the primitives exchanged all devices simultaneously transmit the transcript, which is verified by the group. Intuitively, with more verifiers, it becomes increasingly harder for an attacker to carry our simultaneous signal manipulation attacks, as there are more constraints imposed by the geometry and physical propagation laws. Contrary to prior methods, we showed that the introduction of more devices improves security rather than weakening it.

In our security analysis for all the three scenarios, we always assumed a worst-case attacker where the channels to the legitimate devices are ideal and known to the adversary. These assumptions are different from information-theoretic approaches that rely on a channel advantage to realize secrecy gains.

1.1.2 Aircraft Navigation Verification

The International Air Transport Association has forecasted that over 7.8 billion passengers to travel by 2036 [42]. To cope with the anticipated increase in air traffic, the relevant governing bodies around the world have agreed to NextGen air traffic control technology that shifts traffic surveillance from the uncooperative and independent radar system to a cooperative and dependent digital one [43]. At the heart of NextGen lies the Automatic Dependent Surveillance-Broadcast (ADS-B) standard. In ADS-B, aircraft independently determine their navigation information (location, airspeed, heading, etc.) using onboard satellite equipment (GPS) [13]. To facilitate air traffic management, this navigation information is broadcasted to nearby aircraft and ground air traffic control (ATC) centers. Despite its critical function, ADS-B does not integrate strong security mechanisms. This is because a timely agreement to a common security management framework is an impossible task. Over 190 aviation federations need to manage security across continent and country borders. In NextGen, the aircraft's location is verified by ground stations using a multilateration technique. In this technique, three or more ground stations compute the time difference of arrival from an ADS-B broadcast to validate the claimed aircraft's position [13]. However, an aircraft has no way of verifying the ADS-B broadcast of another aircraft. The ADS-B security vulnerabilities can be abstracted to classical cryptography problems for which solutions are readily available. However, implementing cryptographic solutions on a global scale requires coordination between multiple governing agencies, administrators and operators. Key management operations including key establishment, key refresh, key revocation, certificate management, etc. introduce a substantial layer of complexity and cost to the ADS-B standard.

To cope with these challenges, we examined non-cryptographic solutions for verifying the navigation information broadcasted in ADS-B. We proposed to exploit the Doppler spread phenomenon observed on the wireless channel due to the aircraft motion to verify the claimed aircraft speed, location and heading. The Doppler spread extracted from the channel state information directly translates to the relative radial velocity between the claimer aircraft and the verifier (another aircraft or single ATC). We proposed to use the speed, location and heading of the claimer aircraft estimated from the relative radial velocity using kinematic equations to verify claimed speed, location and heading in the ADS-B frame. We showed that it is very difficult for a static transmitter (or one that moves at speeds significantly lower than those of an aircraft) to emulate the presence of an aircraft moving at a desired velocity and heading.

1.1.3 Fast and Secure Physical Layer Voting

Distributed wireless networks fundamentally rely on the principle of cooperation. Nodes often share information to coordinate network functions and improve the fault-tolerance of distributed operations. As an example, cooperative spectrum sensing is known to improve the detection of licensed user activity in dynamic spectrum access (DSA) [44]. Data fusion is also widely used in wireless sensor networks (WSNs) for improving the performance of target detection, target tracking, and distributed sensing [45]. For many cooperative functions, binary voting algorithms increase fault-tolerance at relative low cooperation overhead. In binary voting, a community of distributed entities shares binary decisions ("yes" or "no") on a parameter of interest (e.g., channel state, the presence of a target). A combining decision rule is applied to collectively determine the decision outcome. This rule is based on some form of majority voting, plurality or threshold, to achieve the desired level of reliability. Typically, binary votes are casted using a messaging scheme, in which 1-bit votes are carried by individual messages. However, message-based voting incurs a relatively high voting delay. In this work, *we define the voting delay as the time period between the initiation of the voting process with the transmission of the first vote by any of the participants until all votes have been received at the tallier*. The tallying time is not accounted as part of the voting delay. For message-based voting, each 1-bit vote is carried by a packet that contains a PHY layer and MAC layer headers. Moreover, verifying the voter authenticity and protecting the integrity of binary votes via digital signatures and message authentication codes, requires additional packet fields. All additional fields (headers, message authentication codes, digital signature) increase the overall transmission time per vote. Further, voters must sequentially access the shared wireless channel to cast their votes. Most popular channel access protocols include anti-collision mechanisms (e.g., backoff process) that further increase the voting delay to cast multiple votes. For time-critical applications, a high voting delay could be unacceptable [46,47].

To address the poor delay scalability of message-based voting, we proposed a *fast* and *secure* voting scheme that implements voting at the PHY layer. Wireless devices exploit the subcarrier orthogonality in the widely adopted orthogonal frequency division modulation (OFDM), to simultaneously cast their votes to an FC within just a few symbols. To overcome the challenges related to decoding simultaneous transmissions from multiple senders, binary votes are cast by adding energy to designated subcarriers. No transmission of preambles and headers is required, as the receiver does not demodulate the OFDM signal. Simple energy detection suffices. We studied the robustness of the proposed scheme against an external and an internal adversary. The former attempts to modify votes by inserting energy into various subcarriers without knowing the subcarrier allocation. The latter is aware of any group secrets used to assign subcarriers, but not of pairwise secrets. The proposed scheme guarantees the integrity of the voting outcome. In addition, we showed that an active adversary who attempts to modify the casted votes, cannot flip the voting outcome at the FC with overwhelming probability. Also, the adversary cannot inject additional votes at the FC. Further, we proposed to improve voting robustness by incorporating the transmission of multiple OFDM symbols to cast a single vote, thus realizing a repetition code. Since OFDM symbols have a very short duration, a repetition code is still far more efficient than messaging. We analytically evaluated the voting robustness as a function of the relevant system parameters under a secret and an open vote model. We presented a prototype implementation of the voting technique on the NI-USRP 2921 platform.

1.2 Dissertation Organization

The remainder of the dissertation is organized as follows. In Chapter 2, we discuss related works. In Chapter 3, we introduce the helper security paradigm that is adopted in several of our works. We also propose HELP, an in-band trust establishment protocol that is resistant to advanced signal manipulations including signal cancellation. In Chapter 4, we present a modulation agnostic in-band pairing protocol which relies on hard-to-forge physical layer property such as signal propagation laws to guarantee message integrity. In Chapter 5, we present a group device pairing protocol, where we exploit the presence of a minimum number of devices in the group to guarantee message integrity during the pairing process. In Chapter 6, we present a technique to authenticate and verify message integrity of ADS-B frames, exploiting the Doppler shift of the channel. In Chapter 7, we present a fast and secure voting scheme for distributed wireless networks, where multiple nodes are able to simultaneously submit votes on a single OFDM symbol. Finally, Chapter 8 summarizes the contributions of this dissertation.

CHAPTER 2

RELATED WORK

In this chapter, we highlight the basics of trust establishment and review the state-of-the-art of trust-establishment techniques. Trust establishment typically consists of two parts. Authentication or verification of identities and key establishment to further bootstrap security services. This is equivalent to the scenario where Alice meets Bob for the first time in the presence of Mallory, as shown in Figure 2.1. To establish trust, Alice and Bob need to first verify each other's identity (mutually authenticate) and agree on a common secret to secure future communications. Trivially, this can be achieved by exchanging a common password over a private channel. However, this method is not scalable for a large number of devices. In addition, many of these devices lack essential interfaces (keyboard, screen, etc.) to enter a common password. Manufacturers attempted to bypass this challenge by shipping network-enabled devices with default passwords. However, such a practice has led to many security vulnerabilities. For instance, the Mirai botnet attack [8] exploited the same default passwords preloaded to IoT devices such as IP cameras, digital video recorders, etc. and attack the DNS infrastructures across the east coast of the US.

Alternately, trust can be established without depending on pre-shared secrets. When pairing without secrets, authentication can be achieved by a Push Button Configuration (PBC) mechanism [48]. A user pushes a button on the legitimate entities D and A within 120 seconds to initiate a pairing protocol. Here authentication is derived from the fact of co-presence. After authentication, a key exchange protocol [49–52] can be executed to establish a common secret. For instance, the two devices can execute the Diffie-Hellman (DH) key exchange protocol [49]. During the DH key exchange shown in Figure 2.3, D and A select *DH exponents* x_D

Figure 2.1: Trust establishment between Alice meeting Bob for the first time in presence of Mallory.

and x_A, respectively. Further, D and A raise their DH exponents to the power of a primitive root g^{x_D} and g^{x_A}), where g is the primitive root of a large prime number p. Each party then independently computes the *DH primitives* as ($g^{x_D} \mod p$) and ($g^{x_A} \mod p$). The two devices exchange the DH primitives ($g^{x_D} \mod p$) and ($g^{x_A} \mod p$) over the public channel. Each device independently computes the key with access to other's DH primitive and its own DH exponent as $k_{DA} = g^{x_D x_A} \mod p$. A passive adversary eavesdropping the key exchange is unable to compute DH exponents x_D and x_A from DH primitives ($g^{x_D} \mod p$) and ($g^{x_A} \mod p$), respectively due to discrete logarithmic problem [53]. However, public message exchanged over the wireless medium is vulnerable to Man-in-the-Middle (MitM) attacks, which are notoriously difficult to thwart without any prior security associations.

As the message exchange during key establishment takes place over the wireless medium an adversary can launch a MitM attack by performing an overshadowing attack [31, 54]. Let the adversary M compute his own DH primitive ($g^{x_M} \mod p$) by selecting his DH exponent x_M. During the overshadowing attack, the adversary M transmits ($g^{x_M} \mod p$) at a significantly higher power than D's message ($g^{x_D} \mod p$), such that A receives ($g^{x_M} \mod p$). Figure 2.2 shows M performing an overshadowing attack on D's transmission such that A receives M's intended signal. The figure shows legitimate D transmitting symbols $\mathbf{x}$ corresponding to ($g^{x_D} \mod p$) which is received as $\mathbf{y}$ at A. Simultaneously, M transmits symbols $\mathbf{x}'$ corresponding to ($g^{x_M} \mod p$) received as $\mathbf{y}'$ at A. Here, $\mathbf{y}'$ is at significantly higher power such that A decodes it instead of $\mathbf{y}$.

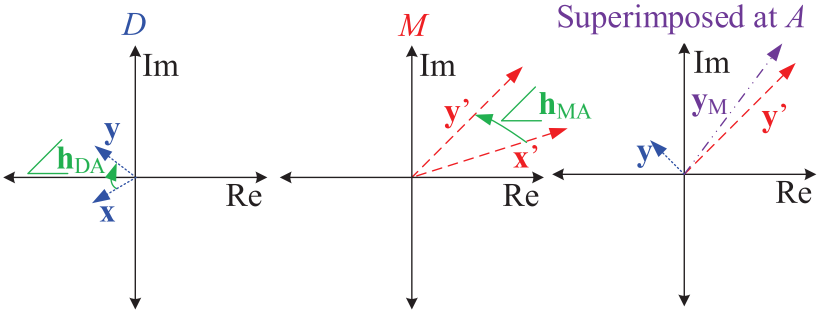


Figure 2.2: Signal overshadowing attack by M on the constellation plane to move the signal transmitted by D to intended signal at A.

D		A
Given $(\mathbb{G}, q, g)$,		Given $(\mathbb{G}, q, g)$,
Pick $x_D \in_U \mathbb{Z}_q$		Pick $x_A \in_U \mathbb{Z}_q$
$m_D \leftarrow g^{x_D} \mod p$	$\xrightarrow{m_D}$	
	$\xleftarrow{m_A}$	$m_A \leftarrow g^{x_A} \mod p$
$k_{DA} \leftarrow (m_A)^{x_D} \mod p$		$k_{DA} \leftarrow (m_D)^{x_A} \mod p$

Figure 2.3: Diffie-Hellman key-agreement on k_{DA} between network-enabled devices D and A.

To thwart MitM attacks, additional message authentication and integrity protection mechanisms are required during the key establishment. Therefore, next, we review the state-of-the-art in authentication/integrity protection without pre-shared secrets. Two classes of authentication protocols are the out-of-band authentication approaches, and in-band authentication approaches.

2.1 Out-of-band Authentication Approaches

To combat a MitM attack, message integrity verification is required during the key exchange of a pairing protocol. For message integrity verification, the out-of-band (OOB) implement a private (other than the radio-frequency (RF)) channel that cannot be accessed by the adversary. However, OOB channels need non-trivial

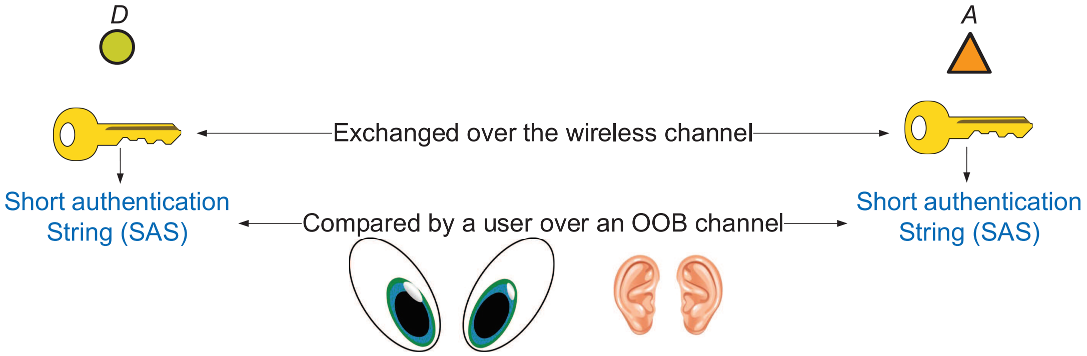

D
A
Exchanged over the wireless channel
Short authentication String (SAS)
Compared by a user over an OOB channel
Short authentication String (SAS)

Figure 2.4: Out of band pairing for the network-enabled devices D and A.

human support and advanced device interfaces. Figure 2.4 shows the general technique for pairing on an OOB channel. After a successful key exchange between D and A. A human user compares a verification string derived from the key established also known as short authentication string (SAS) on an OOB channel. Several secure pairing techniques rely on some OOB channel to defend against MitM attacks [23, 39–41, 55].

Liang *et al.* [39] proposed a technique for pairing a wearable with a computer. The technique guaranteed message integrity by transmitting the hash of the exchanged messages during key establishment over a brightness-modulated visible light channel. This channel requires the availability of a screen and a sensor to detect a change in the levels of brightness. Therefore, this method is not applicable to devices missing these interfaces. Berg *et al.* [55] proposed a pairing method for Bluetooth, in which the user has to perform an action randomly displayed as visual stimuli (action displayed on a display) during the pairing session. Shen *et al.* [40] proposed a commitment-based pairing protocol, where the message integrity during the key established is provided by a user comparing a short authentication string on a visual or verbal channel.

Perkoviĉ *et al.* [23] proposed a group message authentication protocol which utilized a short authentication string (SAS) transmitted over the visual light channel for verification of the messages exchanged over the radio channel. The hash of the exchanged key primitives is displayed by all the devices on the blinking LED channel. Li *et al.* [56] proposed a group device pairing algorithm for body area networks, in which message integrity during the key exchange is provided by matching a short authentication string on flashing LEDs. The protocol supported fast batch deployment, addition and revocation of sensor devices, and did not rely on any additional hardware device. The authentication protocol proposed by Nyang *et al.* [41] depended on a helper device to read a one-time password through a quick response (QR) code. This protocol required the device to have a visual interface like a screen. A cryptographic key distribution algorithm [57, 58] proposed placing legitimate devices attempting to pair inside a Faraday cage. Such that the message exchanged during the pairing protocol is secure and guaranteed message integrity. As an adversary from outside cannot modify the wireless signals inside a Faraday cage. He *et al.* [59] proposed a trust establishment algorithm for WBANs. A trusted third party performed key management such as key establishment, and revocations for all the participating legitimate devices. The third-party management might not be available during the pairing sessions of network-enabled devices.

Biometric based pairing: Alternatively, information such as biometrics can be utilized for pairing which is not accessible for an adversary. These techniques exploit the fact that the legitimate devices placed on the same body record the same biometrics information. This biometric information can be fingerprinted to derive a symmetric key by each device independently. Further, these techniques utilized some error correction such that each device derived the same key even with some error in reading the biometric information. Zheng *et al.* [60] and Karimian *et al.* [61] utilized the fingerprint of the electrocardiogram signal to generate a key on multiple devices within the same body area network. Trust establishment between implantable medical devices and the remote control was proposed [62, 63] utilizing fingerprinting of biometrics for generating a symmetric key. Xu *et al.* [64] proposed symmetric key generation during a pairing between multiple devices on the same body utilizing the characteristics of the gait of a user. However, their applications are

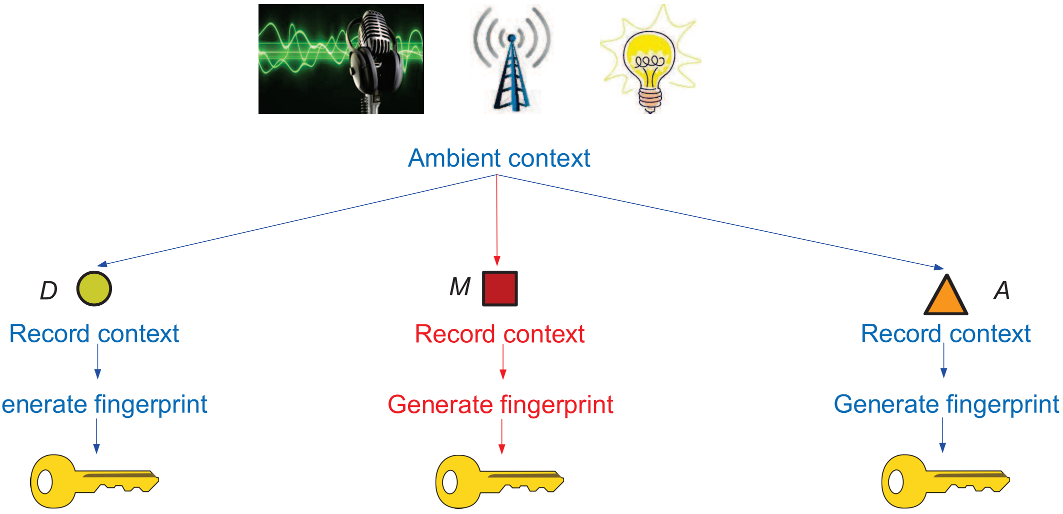


Figure 2.5: Attack by M on ambient context based key establishment between the network-enabled devices D and A.

limited to wearable devices, require uniform sensing hardware, and are susceptible to remote biometrics sensing attacks [65]. However, these protocols are limited to network-enabled devices deployed on a body. In this dissertation, we target to propose protocols that can be implemented for all network-enabled devices.

Ambient context based pairing: Others have exploited the shared physical context for authentication and key agreement. These techniques exploit the fact that the legitimate devices placed in proximity record the same ambient context information. This context information can be fingerprinted to derive a symmetric key by each device independently. Further, these techniques utilized some error correction such that each device derived the same key even with some error in reading the context information. Examples of common modalities include received signal strength (RSS) at both the devices pairing [66] or [67] proposed a pairing method where devices derive a symmetric key using ambient audio. Hayashi *et al.* [68] proposed a probabilistic framework for dynamically selecting an ambient context to derive a symmetric key that satisfied authentication and integrity verification during pairing. Miettinen *et al.* [69, 70] proposed a zero-interaction based pairing protocol. This protocol presents a key establishment and evolution scheme based on fingerprinting of the ambient factors. However, the time for the key establishment depends on the entropy of the ambient factor. The legitimate devices utilizing these techniques require additional and the same hardware to sense the same context. This makes these techniques interoperable. Further, these techniques are vulnerable to an adversary who has access to the common context or can predict a low entropy context information which the legitimate devices use to derive the symmetric key. Figure 2.5 shows such an attack where the adversary can derive the same symmetric key as the legitimate entities with access to the same ambient context as the legitimate entities.

2.2 In-band Authentication Approaches

As an alternative, non-cryptographic authentication techniques usually derive trust from *hard-to-forge* physical-layer characteristics unique to each device/link. They usually transmit information “in-band” without requiring an OOB channel. Existing approaches on non-cryptographic device authentication [71–77] can be classified into three categories: (a) *device proximity*, (b) *location distinction*, and (c) *device identification*.

Device proximity: A proximity-based pairing between devices is based on the assumption, that only the legitimate devices have access to a physically restricted area. All these methods assume that the adversary does not have access to this restricted area. Thus this technique derives message authentication from the proximity. Zheng *et al.* [71] proposed proximity-based pairing method which exploits the channel reciprocity. The legitimate devices participating in pairing derive key by fingerprinting the channel state information (CSI) and received signal strength (RSS). The adversary who is not in the proximity of the legitimate devices is unable to derive the same fingerprint and the key as the legitimate devices. Cai *et al.* [72] utilized multiple antennas on a receiver to verify proximity using the propagation characteristics between the legitimate receivers. The proposed scheme is based on the propagation characteristic of the wireless signal that the power of the received signal is inversely proportional to some exponent of the distance between the sender and receiver. When a nearby sender moved very close to one antenna on the receiver, the receiver observed a large difference between the signal strength measured on its two antennas, whereas a faraway sender would be unable to induce such a large difference. Pierson *et al.* [76] refined the previous idea to be implemented for the commercial-off-the-shelf devices and used a specialized device with two antennas separated by 7cm to measure radio signal strength difference from a device to verify proximity to another device. Some proximity-based authentication [74, 75] utilized fingerprinting the RSS characteristics of an ambient radio source by the legitimate devices in proximity to derive a symmetric key. Zhang *et al.* [77] exploited rapid RSS variation due to wave polarization for detecting proximity between the pairing devices. However, some of these techniques typically require advanced hardware which is not suitable for constrained wireless devices. For example, [76, 78] require multiple-antennas, and [75] needs a wide-band receiver. Moreover, these techniques only address the common key extraction problem, leaving them vulnerable to MitM attacks. Distance bounding techniques [79–81] were also proposed to ensure proximity, but they are not so practical yet (either resort to an OOB channel or a specially designed hardware).

Location distinction: Pairing-based on location distinction assumes that only the legitimate entities have access to certain locations which is inaccessible to an adversary. Exploiting this fact, this technique provides message authentication during key establishment. Varshavsky *et al.* presented amigo [73], where co-located devices derived a symmetric key. The pairing scheme works as follows. First, two devices brought to close proximity, perform an unauthenticated DH key exchange over a wireless radio channel (Wi-Fi). Second, both devices start monitoring the ambient radio environment for a short period of time and construct a signature containing identifiers and signal strength of the packets received during the snapshot. Finally, two devices exchange their signatures over a secure channel using a commitment scheme in order to verify if the received and local measurements match. Location distinction methods such as temporal link signatures that detect location differences [82] require high bandwidth (> 40MHz) and training, which is not always available to low-cost, resource-constrained devices. Ma *et al.* [83] presented two location-aware defense mechanisms for enhanced RFID security and privacy. The protocol used the location information to design selective unlocking mechanisms so that RFIDs can selectively respond to reader interrogations. However, these techniques either require historical PHY-layer data or location information from an OOB channel.

Device identification: The device identification techniques are based on the assumption that there is an initial trust established between the legitimate entities and they know mutual identifiers. This is used for device authentication. Nguyen *et al.* [84] proposed a technique which used a novel self-learning approach to classify devices into device types and builds normal communication profiles for each of these that can subsequently be used to detect anomalous deviations in communication patterns. The proposed protocol utilized a federated learning approach for aggregating behavior profiles efficiently. Robyns *et al.* [85] proposed method to fingerprint the radio signals of LoRa transmitter. This technique utilized machine learning and zero-shot image classification. Unfortunately, both location distinction and device identification techniques require prior training or frequent retraining, which is not applicable to devices first introduced to an environment. A physical unclonable function (PUF) that has been extensively used by researchers to propose solutions for re-authentication of devices [86–88]. In PUF, semiconductor devices receive unique characteristics, which can be recognized during manufacturing. The proposed solutions work very well in producing a unique digital fingerprint of devices. However, these techniques are orthogonal to the initial trust establishment which is the main focus of this dissertation.

Alternatively, in the in-band approach, the pairing process takes place over the RF channel. The key establishment between the network-enabled devices D and A can be achieved using a key agreement protocol such as DH [49]. The DH protocol

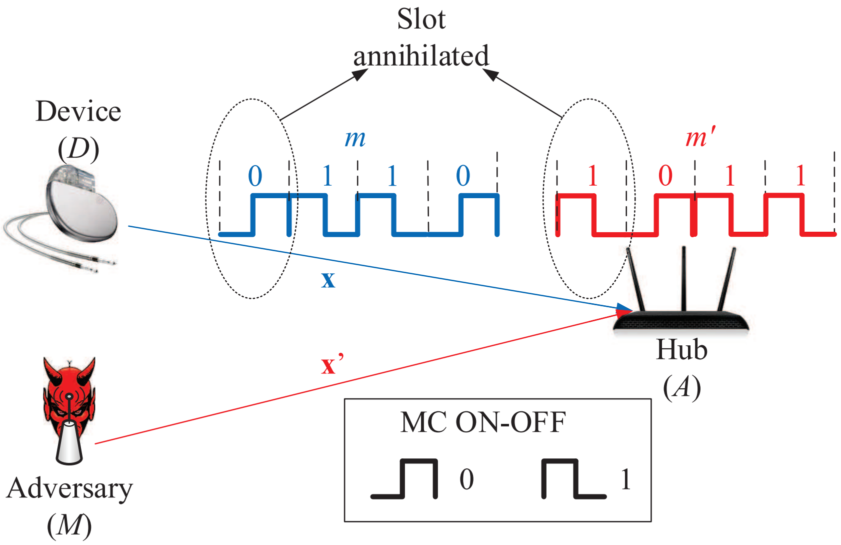


Figure 2.6: D transmits an MC ON-OFF message x to A in the presence of M. To modify x to x', M has to annihilate ON slots of D's transmission.

is proven to be secure against passive adversaries. However, it is still vulnerable to the MitM attacks when DH key primitives are transmitted over wireless.

To preserve the message integrity during a key agreement and prevent MitM attack, several prior works proposed techniques in which the messages during the key establishment are encoded using Manchester-coded ON-OFF keying (MC ON-OFF) [31,33–35], as shown in Figure 2.6. In MC ON-OFF keying, a zero bit is represented with an OFF-ON signal sequence over two slots, whereas the one bit is represented by an ON-OFF signal. ON slots are realized by transmitting random symbols from the constellation plane, whereas OFF slots are realized by no transmission. When using the MC ON-OFF keyed messages, the legitimate receiver or A is expecting ON and OFF slots during each bit. However, when M is performing overshadowing attack to inject m' simultaneously as D is transmitting m, A receives two ON slots in sequence on the bits where $m \neq m'$.

For example, Tamper-Evident Pairing (TEP) proposed by Gollakota *et al.* [33], and integrity codes (I-codes) proposed by Čapkun *et al.* [31] used MC ON-OFF keying during key establishment for detecting overshadowing attacks. However, both assumed the infeasibility of signal cancellation. Based on message integrity, message authentication can be achieved by assuming the presence of the legitimate device is known (a.k.a. authentication through presence). The infeasibility of signal cancellation assumption does not always hold. Pöpper *et al.* demonstrated an effective relay signal cancellation attack using a pair of directional antennas, which works regardless of the packet content and modulation [14]. Recently, Hou *et al.* [34,35] showed that the success probability of signal cancellation attack in the one-to-one setting depends on the randomness of the legitimate channel. A typical indoor environment may not be sufficient because the devices are static and the channel is usually stable. However, these protocols become vulnerable to active MitM attacks, when cancellation of signal [14] becomes possible.

2.3 MitM Over Wireless - Cancellation Attack

Now, we discuss how the MitM attack on the in-band pairing technique can be realized by launching a cancellation attack. A MitM adversary attempting to replace m with m' has to completely annihilate the ON slots of m on those bit positions that the two messages differ. This is generally difficult to achieve under a rich scattering environment due to the unpredictability of the wireless channel between the legitimate parties. At the same time, device authentication is achieved via the verification of co-presence when the user interacts with the devices. In Figure 2.7, we show the cancellation of D's transmission on the constellation plane by M as received by A.

To perform an MitM attack, the adversary has to replace m with m'. Let $\mathbf{x} = \{x(1), x(2), \ldots, x(k)\}$ denote the transmitted symbols modulating m and $\mathbf{y} = \{y(1), y(2), \ldots, y(k)\}$ the received symbols at A. Then,

$$\mathbf{y} = \mathbf{h}_{DA}\mathbf{x}, \tag{2.1}$$

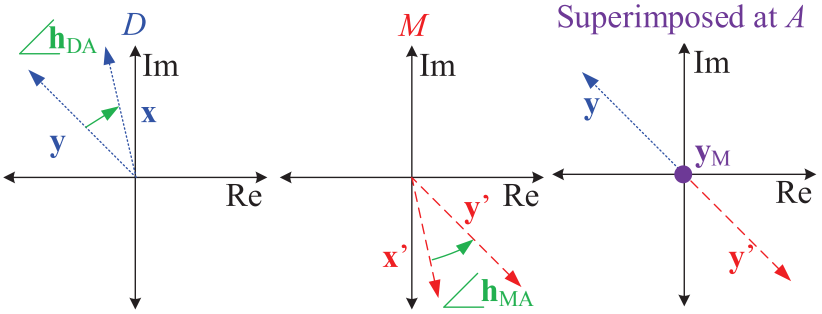


Figure 2.7: Signal cancellation attack by M on the constellation plane to reduce the signal transmitted by D to the noise floor at A.

where $\mathbf{h}_{DA} = \alpha_{DA} \cdot e^{j\phi_{DA}}$ is the impulse response of the D-A channel, α_{DA} is the channel attenuation factor, and ϕ_{DA} is the channel's phase shift. Here, we have assumed that the entire transmission of $\mathbf{x}$ completes within the channel's coherence time, so the channel remains constant. To modify $\mathbf{y}$, the adversary M must transmit $\mathbf{x}'$, modified by the M-A channel to $\mathbf{y}' = \mathbf{h}_{MA}\mathbf{x}'$ such that the superposition $\mathbf{y}_M = \mathbf{y} + \mathbf{y}'$ decodes to m'. In other words, M must compute

$$\mathbf{x}' = \frac{1}{\mathbf{h}_{MA}}(\mathbf{y}_M - \mathbf{h}_{DA}\mathbf{x}), \tag{2.2}$$

and transmit $\mathbf{x}'$ in a timely fashion such that $\mathbf{y}$ and $\mathbf{y}'$ are superimposed as shown in Figure 2.7. According to equation (2.2), the computation of $\mathbf{x}'$ requires the knowledge of the signal $\mathbf{x}$ transmitted by D and of the channels $\mathbf{h}_{DA}$ and $\mathbf{h}_{MA}$. Moreover, the reception of $\mathbf{y}'$ must be synchronized with the reception of $\mathbf{y}$ such that $\mathbf{y}'$ arrives at A within an acceptable delay spread τ_A for correct symbol superposition [89]. Synchronization can be achieved using the preambles or the pilot symbols from the device; such methods are discussed in detail in [90]. The delay spread requirement imposes an important physical constraint on M's locations. The difference between

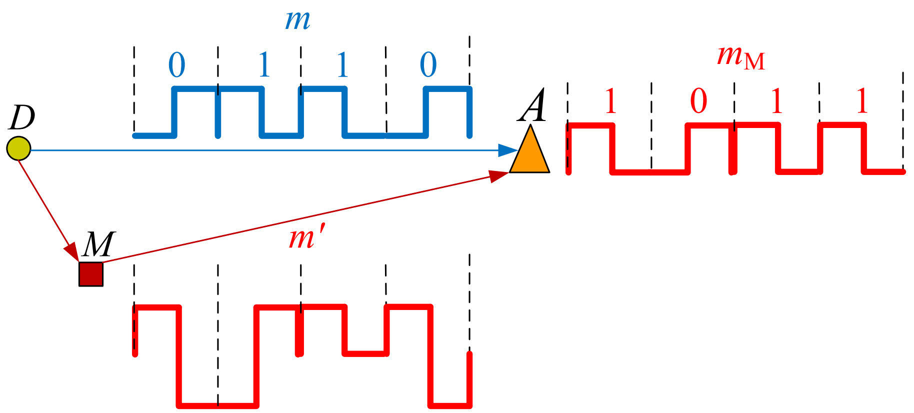


Figure 2.8: MitM attack by M to modify D's MC ON-OFF transmission such that A decodes m_M instead of m.

the adversary's path, and the direct path must satisfy

$$d_{DM} + d_{MA} - d_{DA} \leq \tau_A \cdot c, \tag{2.3}$$

where d_{XY} denotes the distance between X and Y and c is the speed of light.

When the signal $\mathbf{x}$ is MC ON-OFF encoded, denoted by $[\mathbf{x}]$, modification of the received signal to $\mathbf{y}_M$ requires some ON slots of $[\mathbf{x}]$ to be annihilated, i.e., the amplitude of $\mathbf{y}_M$ must be below the signal detection threshold (typically 10s of dBms below zero) in some slots. Practically, obtaining $\mathbf{x}$ in advance to compute $\mathbf{x}'$ is not possible. This is because D can transmit random symbols to implement an ON slot when ON-OFF keying is used. These symbols do not need to belong to a particular modulation mode such as BPSK, QPSK, etc. Alternatively, the adversary can avoid the requirement of knowing $\mathbf{x}$, by performing a relay attack. For performing the relay attack the adversary has to process the signal before relaying such that the received signal at A is exactly opposite phase and same amplitude as that of the signal received from D, as shown in Figure 2.8. To achieve that M with the channel knowledge of $\mathbf{h}_{DA}$, $\mathbf{h}_{MA}$, and $\mathbf{h}_{DM}$, the adversary can compute a phasor and an amplification factor to process the relayed signal. However, since this is an online attack, it is favorable for M to reduce the processing overhead by just performing

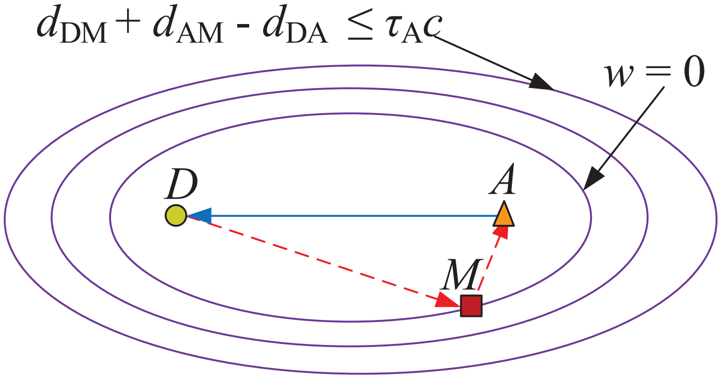


Figure 2.9: To perform signal cancellation, the adversary is placed on an ellipse, centered at D and A that satisfies a path difference of $(2w + 1)\lambda/2$ and does not violate the maximum delay spread τ_A.

the amplification before relaying. For this, the adversary's position is strategically selected such that the path difference between the direct path and the adversary's path satisfies:

$$d_{DM} + d_{MA} - d_{DA} = (2w + 1)\frac{\lambda}{2}, \quad w = 0, 1, 2, \ldots \tag{2.4}$$

where λ denotes the wavelength. This guarantees that the inverse of $\mathbf{y}$ will be received at A when the incoming signal at M is compensated for the respective channel attenuation factors. Because the path difference is an odd multiple of $\lambda/2$, $\mathbf{y}$ and $\mathbf{y}'$ arrive at A with opposite phases, thus canceling each other ($\mathbf{y}_M = 0$). The signal superposition at A for a cancellation attack is shown in Figure 2.7.

We now examine the candidate set of M's locations that lead to successful cancellation via relaying. The adversary's location ℓ_M must satisfy the phase difference equation in (2.4) and the delay spread constraints in (2.3). For (2.4) or (2.4), candidate ℓ_M form a series of ellipses with D and the A placed at the two focal points. The set of such ellipses is shown in Figure 2.9 and is computed by considering all odd integer values of w in (2.4). Finally, the delay spread constraint (2.3) upper

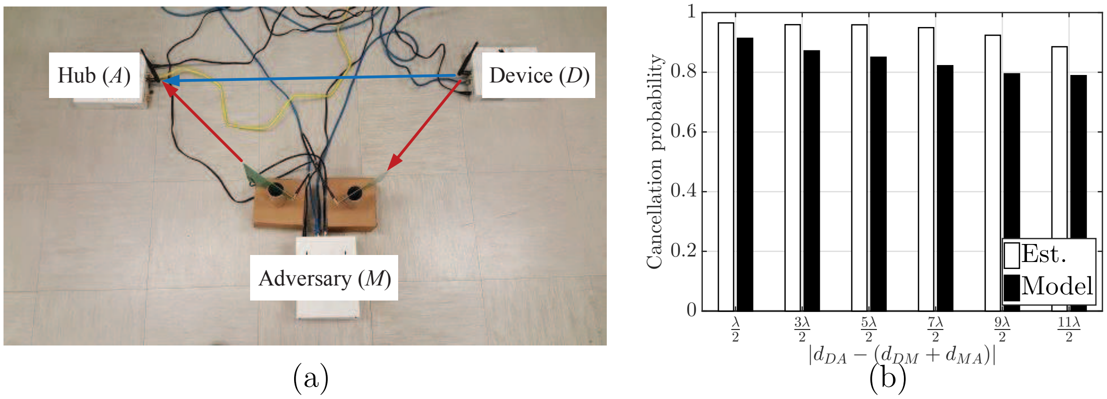


Figure 2.10: (a) Experimental setup for evaluating signal cancellation for a single device-hub pair, and (b) cancellation probability as a function of the distance difference between the direct path from D-to-A and the relay path through M.

bounds w.

To verify the adversary's ability to manipulate signal over wireless, we performed a signal cancellation and injection experiment using four NI-USRP 2921 devices, organized as shown in Figure 2.10(a). All devices were synchronized and transmitted at 2.4GHz. D transmitted a random BPSK-modulated signals during ON slots and no signal during OFF slots to the hub while the adversary M performed relay signal cancellation attack. M is equipped with two USRP devices each having directional antennas (LP0965 Log Periodic PCB Antenna, 850MHz to 6.5GHz), one for receiving symbols from D and other to transmit symbols to A. M is placed such that the path difference between the paths D-to-A and D-to-M-to-A is $\ell \cdot {}^{\lambda}\!/_{2}$ such that the symbols relayed through M are phase shifted by π, to achieve signal cancellation. M used two different techniques for computing the amplitude of the relayed signal, either estimating channels or using LoS far-field channel model [91].

Figure 2.10(b) shows the cancellation probability of the ON slots achieved by the adversary as a function of the difference between the length of the direct path from D-to-A and the relay path through the adversary. We observe that when the adversary is close and therefore, has dominant LoS (e.g., the difference between the path lengths is ${}^{\lambda}\!/_{2} = 6.25$cm), the success probability of symbol modification by relay attack is quite high (92.53% and 86.49% for channel estimation and channel modeling approaches, respectively). This motivates us to investigate a pairing protocol that is resistant to active signal manipulations.

Thus, it remains an open problem as to whether secure in-band device pairing protocols can still be designed under a strong Dolev-Yao attacker which can annihilate wireless signals. In this dissertation, we propose various secret-free techniques, that provide message integrity protection during key establishment and are resilient to advanced signal cancellation attacks.

CHAPTER 3

HELP: HELPER-ENABLED IN-BAND DEVICE PAIRING RESISTANT AGAINST SIGNAL CANCELLATION

3.1 Introduction

3.1.1 Motivation

Recent works have proposed secure device pairing methods that do not rely on pre-shared secrets [20–32,34,35,92]. Most rely on out-of-band (OOB) human verification to provide authentication and verify the protocol success. Human-dependent solutions scale poorly with the number of devices. Some in-band solutions have also appeared, but they almost unanimously derive security from the *infeasibility of advanced wireless signal manipulations, signal cancellation in particular.* To preserve the message integrity during the execution of a key agreement protocol, messages are encoded using Manchester-coded ON-OFF keying (MC ON-OFF). A Man-in-the-Middle (MitM) adversary attempting to replace m with m' has to completely annihilate the ON slots of m on those bit positions that the two messages differ. This is generally difficult to achieve under a rich scattering environment due to the unpredictability of the wireless channel between the legitimate parties. At the same time, device authentication is achieved via the verification of co-presence when the user interacts with the devices. However, as demonstrated by Popper *et al.* [14], this assumption may not hold in many cases. Thus, it remains an open problem as to whether secure in-band device pairing protocols can still be designed under a strong Dolev-Yao attacker which can annihilate wireless signals.

In this chapter, for the first time, we seek an answer to the above question.

Instead of trying to *prevent* signal cancellation attacks, we propose an approach to *detect* the presence of an attacker who attempts to nullify the signal at a receiver. Our core idea for verifying the integrity of a message m is to superimpose another signal from a *helper* device (e.g., a smartphone) while m is being transmitted. Any cancellation attack on m is bound to also cancel the superimposed signal from the helper. The helper is assumed to have an existing trust association with one of the devices in the network (e.g., the hub), and is co-present with the primary device that is authenticated by the hub. The superimposed signal is later revealed by the helper via the authenticated channel, to allow for the recovery of m. Our protocol achieves a strong "tamper-evidence" property where there are no restrictions on what kind of signal manipulation the attacker is allowed to do.

Specifically, the device's message m is encoded with ON-OFF keying and Manchester-coding. During the transmission of m, the helper synchronously injects some random signal at randomly selected slots. Any signal nullification attempt will cancel both the legitimate transmitter's and the helper's signal, presuming that the activity periods for the helper are not easily discernible. The helper later reveals its activity periods via an authenticated channel to enable the hub in the detection of signal nullification attempts. Trust between the hub and the helper is established using traditional means (e.g., input a shared random password on the smartphone when it is first paired with the hub), which is a one-time cost. With only one helper in a network, we can securely introduce many new devices at no extra hardware cost, thus ensuring scalability and usability. Essentially, by exploiting the co-presence of the helper with the new device(s), our protocol transfers the trust from the helper to the new device(s).

3.1.2 Main Contributions and Chapter Organization

The main contributions of this chapter are four-fold:

- We construct a novel physical layer message integrity verification primitive to detect signal cancellation attacks over the wireless channel. We show that our primitive achieves message integrity protection with only in-band communications.

- We utilize the proposed message integrity verification primitive to construct a secure in-band device pairing protocol named HELP based on the Diffie-Hellman (DH) key agreement [49]. Whereas the primitive provides one-way integrity verification (device-to-hub), we show that HELP achieves two-way authenticated key agreement (counter-intuitively). This is done via a novel way that exploits the helper's superposed random signals to simultaneously protect both the integrity and confidentiality of the DH public parameters, such that an adversary impersonating the hub cannot successfully establish a key with a legitimate device.

- We theoretically analyze the security of the proposed integrity verification primitive and the HELP protocol, and we establish bounds for the adversary's success probability under active attacks (especially Man-in-the-Middle attacks). We show that the adversary's success probability is a negligible function of the protocol parameters and thus can be driven to an arbitrary small value.

- We carry out extensive experiments to evaluate the effectiveness of the signal cancellation detection mechanism and the pairing protocol. Our experiments verify that device co-presence significantly hardens the adversary's ability to distinguish between the helper's and the legitimate device's transmissions. We also implement the proposed protocol in our Universal Software Radio Peripheral (USRP) testbed and evaluate the adversary's successful pairing probability with and without the protection of our integrity verification primitive. The experimental results are in line with our analytical findings.

Chapter Organization: The remainder of this chapter is organized as follows. We state the system and threat models in Section 3.2. We present the integrity verification primitive and the HELP pairing protocol in Section 3.3. The security of the pairing primitive and of HELP are analyzed in Section 3.4. In Section 3.5, we study the adversary's capability in inferring the helper's transmissions and injecting modified messages by performing experiments on the USRP platform. We further experimentally evaluate the HELP assisted key-agreement protocol. We conclude the chapter in Section 3.6.

3.2 Model Assumptions

3.2.1 System Model

We consider a star network topology, where a wireless hub (A) services multiple personal devices, which is similar to an Internet-of-things (IoTs) scenario. For example, the network can reside inside a home or an office space. Our goal is to securely pair an unauthenticated device with the hub in the presence of an adversary and establish a common key between the device and the A. The adversary can either try to hijack the uplink communication to pair with the A, or spoof a rogue A to pair with a legitimate device. The device and A do not pre-share any common secrets (*e.g.* secret cryptographic keys). We assume that a user initiates the pairing process by powering the device and setting it to pairing mode. Figure 3.1 describes the system model. Formally, the following entities are part of the system model.

Hub (A): The A serves all the legitimate devices and needs to establish a secure communication link with each of them. The A connects with the legitimate devices through a wireless channel. The A verifies and pairs with any legitimate device requesting to join the network.

Helper Device (H): The helper is an auxiliary device such as a smartphone, that assists the A in the pairing process. The helper has already established a secure authenticated channel with the A, either by establishing a common key, using a public/private key pair, or through some OOB channel [24, 92]. Using this secure

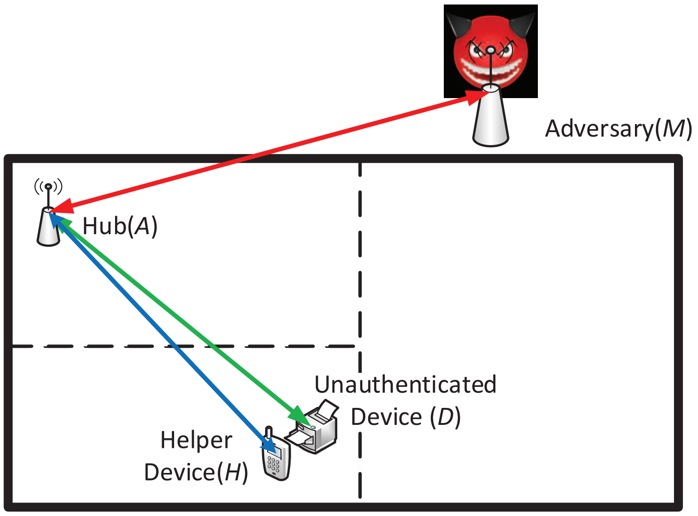


Figure 3.1: Entities of the system model and basic setup.

channel, H can apply an authenticated encryption function AE($\cdot$) on a message m_H to guarantee the confidentiality and integrity of m_H, and the authenticity of the source. Any such AE($\cdot$) can be utilized with the proposed protocol. For example, if H and the A share a public/private key pair, H can encrypt/sign/encrypt (or sign/encrypt/sign) its message to guarantee the necessary security properties. If H and D share a common master symmetric key, an encrypt-then-MAC operation can be followed to implement AE($\cdot$), after separate symmetric keys are generated from the master key for the encryption and MAC operations. One of the examples is to use encryption then message authentication code hashing with the shared key. We refer the reader to [93] for more details on authenticated encryption. We leave the exact specification of AE($\cdot$) open to allow for both symmetric and/or asymmetric methods.

Note that pairing H to the A is a one-time effort and need not be repeated with every device join. Moreover, only the helper is required to have an advanced interface to pair with the A.

Legitimate Device (D): A legitimate device is a typical network-enabled device which does not share any secrets with the A or H. The device is usually small and has simple user interfaces (such as a power button) and hardware capabilities. The legitimate device, H, and the A are assumed to be co-present during the pairing process. H and D are placed in close proximity such that they have a highly correlated wireless channel.

3.2.2 Threat Model

Adversary: We consider the typical *Dolev-Yao model* [94]. The adversary (M), can fully control the wireless channels of the network. For example, it can eavesdrop, modify, remove, replay or inject messages (frames) transmitted on the wireless channel. The adversary is also powerful enough to annihilate signals transmitted from D and H over the wireless channel, such that they do not reach the A (and vice versa). This can be accomplished by techniques proposed by Pöpper *et al.* [14]. The pairing protocol itself is known to M, but the adversary does not have physical access to any of the devices. The helper device is assumed to be trusted and its secret key with the A is kept away from adversaries.

Note that we do not impose any location restriction for the attacker. Although the devices are typically located in a physically bounded area such as a home, we do not assume that this is a secure region. Instead, the attacker can be located inside the physical space, as long as the attacker cannot physically control the device and the A to be paired. That is, the attacker does not control the helper so that it cannot initiate the pairing with the A when no legitimate device is present. The user is aware of the presence of both the A and of the legitimate device (which are powered on) when the pairing is initiated. This is the minimal assumption adopted by the majority of the previous works in device pairing.

The goal of an attacker is to pair successfully with the A and/or D. Therefore, we mainly consider a MitM attacker in our security analysis. However, in this chapter, we do not focus on preventing denial-of-service (DoS) attacks such as jamming, which is orthogonal to our studies. Similarly with all relevant literature, we assume that the adversary is incapable of physically blocking signals (e.g., by adding a Faraday cage) to the device, the helper, or the hub.

In addition, at any point in time, the attacker may try to find out who is transmitting on the wireless channel. There could be several cases: device only, helper only, A only, or device plus helper together. For example, the attacker can do so via energy detection or use physical layer identification/fingerprinting techniques [95–100]. Since we assume that D and H have a highly correlated channel due to their proximity, it is generally difficult for the attacker to differentiate between the cases of device only and helper only. Thus, the attacker can differentiate between the number of transmitters (i.e., $D + H$ or D/H alone), but the attacker cannot perfectly distinguish D and H (i.e., the probability of successful detection is less than 100%). We propose specific power and slot synchronization randomization methods to ensure that D and H are not easily distinguishable. Note that any device distinction method has to operate only to correspond to the online nature of a MitM attack.

3.3 HELP: Helper-enabled Pairing

In this section, we present HELP, an in-band Helper-enabled pairing protocol that does not require secret preloading. HELP makes use of a new PHY-layer message integrity protection primitive to detect signal cancellation attacks that are launched to perform a MitM attack against a key agreement protocol. We first describe the PHY-layer protection primitive and then use this primitive to construct HELP.

3.3.1 Message Integrity Protection Against Signal Cancellation

Consider the simple scenario depicted in Figure 3.1. A new legitimate device D wants to pair with the A by transmitting a message m_D over a wireless channel. Message m_D is not protected by any cryptographic message integrity mechanism such as a MAC because D and the A do not share any prior security association. Let $\mathbf{x}_D$ denote the corresponding signal transmitted from D carrying m_D.

Let also an adversary M perform a signal cancellation attack on the received signal $\mathbf{y}_D = \mathbf{h}_{D,A}\mathbf{x}_D$ at the A, where $\mathbf{h}_{D,A}$ denotes the channel between D and the A. Simultaneously, M injects his own signal $\mathbf{x}_M$ carrying message m_M. The main challenge in providing message integrity is to detect that a cancellation/injection has taken place.

To combat signal cancellations, we employ Manchester-coded ON-OFF (MC ON-OFF) keying modulation to transmit m_D from D to the A similar to [31, 33]. In ON-OFF keying, a zero bit is mapped to (OFF, ON) slots pair, whereas a one bit is mapped to (ON, OFF) slots pair. The receiver demodulates the ON-OFF keying sequence by applying energy detection on every slot. The advantage of ON-OFF keying is that it hardens signal cancellations, as the adversarial device, M has to "erase" the received signal $\mathbf{y}_D$ at the A by synchronizing its own signal transmission $\mathbf{x}_M$ and taking into account the channels $\mathbf{h}_{D,A}$ and $\mathbf{h}_{M,A}$. Different from previous approaches [31,33,101], we consider the worst case scenario where signal cancellation is possible due to the stability and predictability of the respective channels, as it was demonstrated in [14].

The MC ON-OFF keying facilitates several functions. First, the alteration between ON and OFF slots prevents the zero wandering problem, allowing the receiver to keep a power reference for differentiating between ON and OFF slots, irrespective of the data sequence. More importantly, an MC ON-OFF message contains an equal number of zeros and ones. Our integrity protection mechanism relies on the detection of canceled ON slots and therefore, the guarantee of ON slots irrespective of the data sequence is critical to the protocol security. Finally, the use of MC ON-OFF keying allows for the recovery of the device's message when the latter has been corrupted from the intentional transmissions of the helper. Revealing the "time locations" of the helper's ON slots enables the message recovery.

In the proposed integrity primitive, the helper is placed in close proximity to the unauthenticated device D and synchronously transmits a message m_H while m_D is being transmitted. A signal cancellation targeted at the A is bound to also cancel

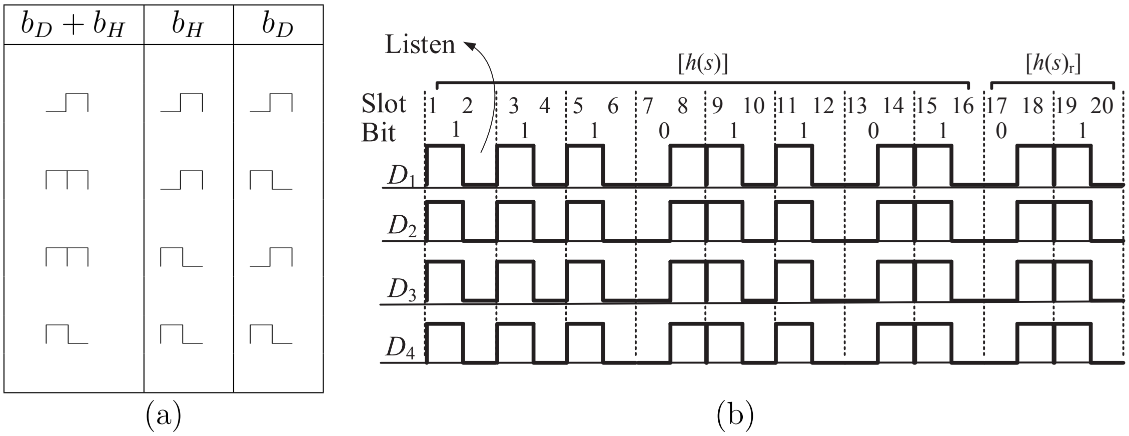


Figure 3.2: (a) Truth table for recovering $[h(m_D)']$ from $([h(m_D)] + m_H)'$, using $\mathbf{s}$, and (b) an example of recovering $[h(m_D)']$ from $([h(m_D)] + m_H)'$.

the signal from H. With the completion of the m_D transmission, the helper reveals m_H to the A, who verifies if any part of m_H has been canceled.

If the message integrity verification test is passed, the A exploits the knowledge of m_H to recover m_D. A key requirement for the successful detection of signal cancellations is that the adversary M cannot *swiftly* identify the ON slots of the helper. We achieve this requirement by placing the helper in close proximity to D and by randomizing the transmit power and the starting time of each ON-OFF slot at D and H. Placing H close to D makes it difficult to differentiate the two devices using transmission directionality or the uniqueness of the wireless channel. Note that the ON-OFF transmissions contain no preambles, so channel estimation becomes difficult. The randomization of the power and ON slot firing times aim at preventing the device distinction using RSS measurements or the possible time misalignment between the two devices due to inaccurate synchronization or different paths to the adversary. We emphasize that any device distinction mechanism must operate online—the adversary has to decide to cancel an ON slot within the first few samples—which renders existing sophisticated radio fingerprinting techniques inadequate [95–100]. We now describe the PHY-layer message integrity verification primitive in detail.

3.3.2 HELP Integrity Verification

We propose a message integrity verification method called HELP that operates with the assistance of a helper device H. The integrity of a message m_D transmitted from D to the A is verified via the following steps.

1. **Device Placement:** The helper H is placed in close proximity to the unauthenticated device D.

2. **Initialization:** The user presses a button on D or simply switches D on to set it to pairing mode. The user then presses a button on H to initiate the protocol. The helper sends an authenticated *request-to-communicate* message to the A using the AE$(\cdot)$ function. This message attests that the legitimate device D is present and H is placed near D.

3. **Device Synchronization:** The A sends a publicly known synchronization frame SYNC to synchronize the clocks of D, H and itself[1]. The SYNC frame is similar in function to the known preamble that is attached to wireless transmissions for synchronizing the receiver to the transmitter. In our protocol, all three entities synchronize to the same time reference, using the known SYNC message.

4. **Transmission of m_D:** D transmits m_D in the form $[h(m_D)], m_D$, where $[\cdot]$ denotes an MC ON-OFF message and h is a cryptographically-secure hash function. Note that no key input is used with h, as D and the A do not share a common key.

5. **Helper Signal Superposition:** Synchronously with the transmission of $[h(m_D)]$, the helper transmits a signal m_H with ON slots in a random number of slot locations determined by vector $\mathbf{s}$. The ON slots in $\mathbf{s}$ are time-aligned

[1] The SYNC message doesn't need to be secured since if it is canceled at both device and helper, it becomes a DoS attack. If the device and helper are forced to be out of sync by an attacker, A will fail to decode which is again a DoS.

with the slots (ON or OFF) of $[h(m_D)]$. Only one slot of m_H can be ON per MC ON-OFF bit of $[h(m_D)]$. Sequence m_H is not necessarily a proper MC ON-OFF sequence (and hence, is not marked by $[\cdot]$).

6. **Reception at the A:** The A receives $([h(m_D)] + m_H)'$ and m'_D.

7. **Revealing m_H:** The helper reveals $\text{AE}(\mathbf{s}, K)$ to the A.

8. **Integrity Verification of $\mathbf{s}$:** The A decrypts $\mathbf{s}$ and verifies its integrity using function $\text{VD}(\cdot)$, which is the corresponding decryption/verification function to $\text{AE}(\cdot)$. If verification fails, the A aborts m'_D.

9. **Integrity Verification of m_D:** The A verifies that all slot locations indicated by $\mathbf{s}$ are ON on the received $([h(m_D)] + m_H)'$. If not, a signal cancellation attack is detected and m'_D is rejected. Otherwise, the A recovers $h(m_D)'$, by removing m_H from $([h(m_D)] + m_H)'$ using the knowledge of $\mathbf{s}$. For bits where $\mathbf{s}$ was OFF in both corresponding slots, the MC ON-OFF sequence is decoded using typical decoding. For an ON slot in $\mathbf{s}$, a bit b_D is decoded using the truth table in Figure 3.2(a). Upon recovery of $h(m_D)'$, the A checks if $h(m'_D) \overset{?}{=} h(m_D)'$. If the integrity verification fails at the A, either the A or H display a FAILURE message, and all entities abort the protocol. The user has to restart the pairing process from the initialization step. If the integrity verification passes, then A or H display a SUCCESS message.

The steps for extracting $[h(m_D)']$ from $([h(m_D)] + m_H)'$ at the A are shown in Figure 3.2(b). After synchronization, D transmits $h(m_D) = 0110110101$ in the form of $[h(m_D)]$ (for illustration purposes, we have restricted the length of the hash function to 10 bits). The helper synchronously transmits during slots $\mathbf{s} = \{4, 10, 13, 15, 18\}$. The A receives the superimposed signal $([h(m_D)] + m_H)'$. Using the truth table in Figure 3.2(a), the original MC ON-OFF sequence corresponding to $h(m_D)$ is recovered.

3.3.3 Device Pairing with HELP

In this section, we describe how the A and D can establish a secret key in the presence of a MitM adversary. We complement the DH key agreement protocol with the HELP integrity verification primitive. The latter is used to detect the cancellation portion of a MitM attack. Moreover, the helper provides the necessary authentication for the DH message exchange. The HELP-enabled DH message exchange is shown in Figure 3.3.

To fix the ideas, the A (or D) publishes parameters $(\mathbb{G}, q, g)$ of the DH scheme, where ($\mathbb{G}$ is a cyclic group of order q and g is a generator of G). If $(\mathbb{G}, q, g)$ are already publicly known, they need not be sent by either party. Device D computes $z_D = g^{X_D}$, where X_D is chosen from $\mathbb{Z}_q$ uniformly at random. After the initialization and synchronization steps (omitted from Figure 3.3), D transmits the integrity-protected form of $m_D : ID_D, z_D$ to the A, while the helper is injecting m_H on slot positions denoted by $\mathbf{s}$. Here, we opt to protect both $h(m_D)$ and m_D with the PHY-layer primitive to conceal the value of m_D from an adversary M, who cannot learn the helper's sequence m_H. This prevents a rogue A from recovering m_D, so that it cannot pair with the device successfully. The helper then reveals $\mathbf{s}$ to the A through the secret channel implemented by AE$(\cdot)$. The A uses $\mathbf{s}$ to verify the integrity of m_D and recover z_D. A replies with $z_A = g^{X_A}$, where X_A is chosen in $\mathbb{Z}_q$ uniformly at random. Each party independently calculates $k_{D,A} = g^{X_D \cdot X_A}$. Immediately following the key-agreement, D and A engage in a key confirmation phase, initiated by D. This can be done by executing a two-way challenge-response protocol [102], as shown in Figure 3.4. If any of the verification steps fail, the corresponding party aborts the pairing protocol.

3.4 Security Analysis

In this section, we analyze the security of the HELP integrity verification primitive and evaluate the security of the DH-based pairing protocol presented in Section 3.3.3.

D		A
Given ID_D,		Given ID_A,
$(\mathbb{G}, q, g)$		$(\mathbb{G}, q, g)$
Pick $X_D \in_U \mathbb{Z}_q$		$X_A \in_U \mathbb{Z}_q$
$z_D \leftarrow g^{X_D}$		$z_A \leftarrow g^{X_A}$
$m_D \leftarrow ID_D, z_D$		$m_A \leftarrow ID_A, z_A$
(H active)	$\xrightarrow{[h(m_D),m_D]+m_H}$	
(H active)	$\xrightarrow{\text{AE}(\mathbf{s},K)}$	Verify
		& Extract z_D
	$\xleftarrow{m_A}$	
$k_{D,A} \leftarrow (z_A)^{X_D}$		$k_{D,A} \leftarrow (z_D)^{X_A}$

Figure 3.3: Diffie-Hellman key-agreement on $k_{D,A}$ using the HELP PHY-layer integrity verification method.

D		A
$C_D \in_U \mathbb{Z}_q$		
	$\xrightarrow{ID_D,C_D}$	
		$R_A \leftarrow h_{k_{D,A}}(ID_A \Vert C_D \Vert 0)$
Ver(R_A)=true?	$\xleftarrow{R_A}$	
		$C_A \in_U \mathbb{Z}_q$
	$\xleftarrow{ID_A,C_A}$	
$R_D \leftarrow h_{k_{D,A}}(ID_D \Vert C_A \Vert 1)$		
	$\xrightarrow{R_D}$	Ver(R_D) = true?

Figure 3.4: Key confirmation of $k_{D,A}$ using a challenge-response protocol.

3.4.1 Security of the HELP Primitive

Consider the transmission of $[h(m_D)], m_D$ from D to the A, superimposed with the transmission of m_H. The goal of the adversary M is to replace m_D with some desired m'_D and pass the verification at the A. In the absence of the helper, a straightforward strategy for M is to annihilate $[h(m_D)], m_D$ and inject $[h(m'_D)], m'_D$. However, when m_H is superimposed on $[h(m_D)]$, a cancellation of $[h(m_D)] + m_H$ leads to the likely detection of the cancellation attack due to the "erasure" of the helper's ON slots.

Rather than blindly canceling the composite signal $[h(m_D)] + m_H$ transmitted by D and H, the adversary can attempt to detect the ON slots of the helper and leave those intact. He can then target only the OFF symbols of m_H and modify those to desired values so that the A decodes m'_D. To pass the integrity verification performed by the A, it must hold that (a) all the ON slots indicated in $\mathbf{s}$ are also ON slots in $[h(m'_D)] + m_H$, and (b) the removal of m_H during step 8 of HELP (see Section 3.3.2), leads to the decoding of $[h(m'_D)]$. As m_D follows in plaintext, the adversary can then replace m_D with m'_D.

We first show that if the adversary can identify the ON slots of the helper (this is equivalent to knowing m_H), then it can modify the transmitted signal such that the desired value m'_D is decoded at the A. Consider the transmission of one MC ON-OFF bit b_D and the superposition of an ON slot by H either during the ON or the OFF slot of the coded b_D. The possible outcomes of this superposition are shown in the third column of Table 3.1. Moreover, we show the signal b_M that must be injected by M to cause the decoding of the desired value b'_D at the A. For illustration purposes, we show the signal cancellation as a negation of the ON value.

From Table 3.1, we observe that if b_H is known, the adversary can always make the A decode the desired bit b'_D, irrespective of the value of b_D. Moreover, since the ON bits of m_H stay intact, the modified signal will pass the PHY-layer integrity verification at the A. However, identifying the ON slots of the helper is difficult due to the location proximity between D and H and also the strict reaction time necessary to perform the cancellation attack in an online fashion. In the next proposition, we prove the security of the integrity verification mechanism under the realistic assumption that an ON slot for the helper is timely identified by M with some probability. We experimentally evaluate this probability in Section 3.5. The security of the integrity verification of HELP is given by Proposition 1.

Proposition 1. *The HELP integrity verification primitive is δ–secure with*

$$\delta = \left(1 - \frac{1-p_I}{4}\right)^{|\mathbf{s}|}. \tag{3.1}$$

Here δ is the probability that the A accepts a message forgery by M, $|\mathbf{s}|$ is the length

Table 3.1: Injection of desired bit b'_D, when the ON slots of the helper can be detected.

	b_D	b_H	$b_D + b_H$	b_M	$b_D + b_H$ $+b_M$	b'_D
1	_⌈⌉	_⌈⌉	_⌈⌉	_⌈⌉	_⌈⌉	_⌈⌉
2	_⌈⌉	_⌈⌉	_⌈⌉	⌈⌉_	⌈⌉⌈⌉	⌈⌉_
3	_⌈⌉	⌈⌉_	⌈⌉⌈⌉	_⌈⌉	⌈⌉⌈⌉	_⌈⌉
4	_⌈⌉	⌈⌉_	⌈⌉⌈⌉	⌈⌉⌊⌋	⌈⌉_	⌈⌉_
5	⌈⌉_	_⌈⌉	⌈⌉⌈⌉	⌊⌋⌈⌉	_⌈⌉	_⌈⌉
6	⌈⌉_	_⌈⌉	⌈⌉⌈⌉	⌈⌉_	⌈⌉⌈⌉	⌈⌉_
7	⌈⌉_	⌈⌉_	⌈⌉_	_⌈⌉	⌈⌉⌈⌉	_⌈⌉
8	⌈⌉_	⌈⌉_	⌈⌉_	⌈⌉_	⌈⌉_	⌈⌉_

of the vector indicating the number of the helper's ON slots, and p_I is the probability of inferring the helper's activity during one MC ON-OFF bit when D and H do not co-transmit. Here, δ is a negligible function of $|\mathbf{s}|$. In eq. (3.1)*, it is assumed that a strongly universal hash function is used as part of the HELP primitive.*

Proof. Assume that the adversary M wants to modify the message m_D sent from D to the A to a message $m'_D \neq m_D$. To accept m'_D, the A must correctly receive $[h(m'_D)], m'_D$ and all the slots indicated in $\mathbf{s}$ must be ON slots. The modification of m_D to m'_D can be made by canceling m_D and injecting m'_D. However, to pass verification, M has to modify $[h(m_D)]$ to $[h(m'_D)]$. Since, m_D is unknown to the adversary while $[h(m_D)]$ is being transmitted due to the one-wayness of $h(\cdot)$, M cannot predict the signal transmitted from D.

To modify $[h(m_D)]$, the adversary must launch a signal cancellation on $[h(m_D)]+ m_H$ and inject $[h(m'_D)]$ at the same time. Moreover, all the ON slots denoted in the helper's location vector $\mathbf{s}$ must remain as ON slots in $[h(m'_D)]$. Also, the A must decode $[h(m'_D)]$ after m_H is removed. This can be achieved if M does not apply any cancellation on the ON slots indicated in $\mathbf{s}$ and modifies the rest of the slots (OFF slots in m_H) to decode to the desired message. The signal injections of M are made according to Table 3.1.

The derivation of the probability δ that the adversary's modification is accepted at the A is performed in two parts. In the first part, we derive the probability that M's cancellation/injection is detected, when M modifies the transmission one bit. We then compute the probability of detecting signal modifications by M over all bits. Consider the i^{th} bit of $h(m'_D)$ which corresponds to Manchester-coded slots t_{2i-1} and t_{2i}.

Here, we assume a probability p_I, which is the probability of inference of detecting the presence of H's signal. This is discussed in details in the Section 3.5. Here we state an assumption, that if H's signal is detected the adversary does not cancel the signal. The probability of cancel is $(1 - p_I)$.

The adversary is detected for i^{th} bit on which H is active, for two conditions with wrong inference $(1 - p_I)$. (a) First, the helper bit is zero *i.e.* H injects energy on t_{2i} slot, device bit is one slot and adversary bit is one. (b) Second, the helper bit is one *i.e.* H injects energy on the t_{2i-1} slot, device bit is zero and the adversary bit is zero.

Let P_r denote the probability that the A rejects the corresponding bit of $[h(m'_D)]$ at bit b_i due to cases (a) and (b). This probability can be calculated as:

$$\begin{aligned} p_r &= (\Pr[b_i^H = 0, b_i^D = 1, b_i^A = 1] \\ &\quad + \Pr[b_i^H = 1, b_i^D = 0, b_i^A = 0]) \\ &\qquad \Pr[\text{wrong inference}] \\ &= \left(\frac{1}{2}\cdot\frac{1}{2}\cdot\frac{1}{2}+\frac{1}{2}\cdot\frac{1}{2}\cdot\frac{1}{2}\right)(1-p_I) \\ &= \frac{1-p_I}{4}, \end{aligned} \tag{3.2}$$

In (3.2), b_i^X denotes the transmitted value of device X at bit b_i, and p_I is the probability of inference of helper's activity by the M on a given bit. For (3.2), we have used the fact that a strictly universal hash function is the part of HELP. For a strictly universal hash function, output hashes for two different inputs differ on each bit with probability 1/2.

The probability δ of accepting the modified message of M at the A is computed by taking into account all $|\mathbf{s}|$ cardinality of the set of bits on which the helper was active. The adversary's modified message is accepted by the A if *none of the bits* in $|\mathbf{s}|$ is rejected. Each bit b_i is rejected with probability p_r given by (3.2). As rejection on each slot occurs independently, the overall probability of accepting $[h(m'_D)]$ is computed via the Binomial distribution with parameter p_r. That is,

$$\begin{aligned}\delta &= 1 - \sum_{x=1}^{|\mathbf{s}|} B\left(x, |\mathbf{s}|, p_r\right) \\ &= 1 - \sum_{x=0}^{|\mathbf{s}|} B\left(x, |\mathbf{s}|, p_r\right) + B\left(0, |\mathbf{s}|, p_r\right) \\ &= (1 - p_r)^{|\mathbf{s}|} \\ &= (1 - \frac{1 - p_I}{4})^{|\mathbf{s}|}. \end{aligned} \tag{3.3}$$

where $B(\alpha, \beta, \gamma)$ is the Binomial probability density function.

We now show that δ is a negligible function of $|\mathbf{s}|$.

In (3.3), δ is a negligible function if $(1-p_r)^{|\mathbf{s}|}$ is shown to be a negligible function. To prove the latter, let $\mu(|\mathbf{s}|) = a^{-|\mathbf{s}|}$ where $a = \frac{1}{1-p_r}$. For $\mu(|\mathbf{s}|)$ to be a negligible function, $\forall\ c \in \mathbb{N}$ there exists a $n_0 \in \mathbb{N}$ such that $|\mathbf{s}| > n_0$ and $\mu(|\mathbf{s}|) < n^{-c}$. Let $n_0 = c^{\frac{a}{a-1}}$. Then

$$\begin{aligned} a^{|\mathbf{s}|} &= \left(a^{\log_a |\mathbf{s}|}\right)^{-\frac{|\mathbf{s}|}{\log_a |\mathbf{s}|}} \\ &= \left(|\mathbf{s}|\right)^{-\frac{|\mathbf{s}|}{\log_a |\mathbf{s}|}}, \end{aligned}$$

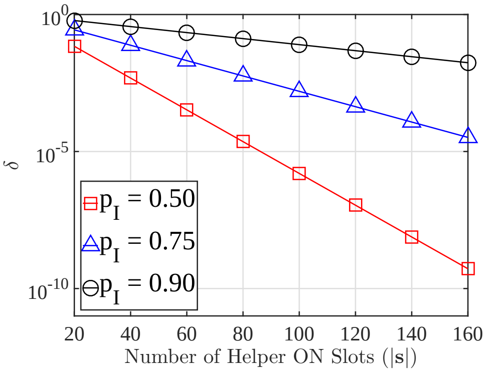


Figure 3.5: Probability of accepting a forged message m'_D at the A as a function of $|\mathbf{s}|$, for varying inference capabilities of helper activity.

Since $|\mathbf{s}| > n_0$, it follows that

$$\frac{|\mathbf{s}|}{\log_a |\mathbf{s}|} > \frac{n_0}{\log_a n_0} > \frac{n_0}{n_0^{\frac{1}{a}}} > c.$$

Therefore,

$$\begin{aligned} \mu(|\mathbf{s}|) &= a^{-|\mathbf{s}|} \\ &= (|\mathbf{s}|)^{-\frac{|\mathbf{s}|}{\log_a |\mathbf{s}|}} \\ &< n^{-c}. \end{aligned}$$

This proves that $(1-p_r)^{|\mathbf{s}|}$ is a negligible function for $a \neq 1$ or equivalently $p_r \neq 0$, thus concluding the proof on the negligibility of δ for $p_r \neq 0$. □

In our analysis, we set the inference probability of H's activity to one when either D and H co-transmit or none transmits. In the former case, the presence of high power can be used to detect the superposition of D and H ON slots, and hence infer H's ON slot. In the latter case, the absence of power can be used to detect a helper's OFF slot. When either D or H are active, the inference probability is set to $p_I < 1$ due to the ambiguity in deciding which of the two devices is active.

Summarizing,

$$\Pr[\text{Inference}] = \begin{cases} 1, & D \; \& \; H \text{ transmit} \\ 1, & D \; \& \; H \text{ do not transmit} \\ p_I, & \text{D or } H \text{ transmits.} \end{cases} \tag{3.4}$$

In Proposition 1, δ depends on two variables; the cardinality of $\mathbf{s}$ and p_I. From (3.1), it is evident that δ is a negligible function of $|\mathbf{s}|$, and a monotonically increasing function of p_I. In Figure 3.5, we show δ as a function of $|\mathbf{s}|$ for various values of p_I. As expected, a higher p_I yields a higher δ value for the adversary. For instance, when $p_I = 0.9$, $\delta = 0.0174$, when $|\mathbf{s}| = 160$, which may not be acceptable. However, doubling the size of $\mathbf{s}$ lowers δ to 0.0003. Note that in a single use of the HELP primitive, the attacker has only one chance to guess $\mathbf{s}$ and modify the value of m_D in an online fashion. Hence, a higher probability of forgery is acceptable here relative to standard cryptographic security (similar security values are sought in previous pairing protocols, which use short authentication strings [27]).

3.4.2 Security of the Device Pairing Protocol

We now analyze the security of the device pairing protocol proposed in Section 3.3.3. Since the security of the DH key-agreement protocol under a passive adversary is standard [103], we focus on the security under active attacks. We divide our analysis into two parts. In the first part, we examine if the adversary can pair a rogue device to a legitimate A. In the second part, we examine if a legitimate device can be deceived to pair with a rogue hub. These two steps are part of a MitM attack.

Pairing a Rogue Device with a legitimate A

The pairing of a rogue device D' with the A can occur under two different scenarios: (a) D' pairs in the absence of a legitimate device D, and (b) D' pairs while D and the A execute a pairing session.

Pairing in the absence of a legitimate device: The pairing protocol described in Section 3.3.3 is initiated with the placement of H in close proximity to the legitimate device and the press of a button on H and D, respectively. The button pressing sends a pairing initialization message to the A which is authenticated using the secure AE$(\cdot)$ function. Without access to the helper device, the adversary cannot initiate the pairing process from a remote location.

Hijacking a legitimate pairing session: Since M cannot initiate the pairing process with the A, he can only attempt to pair a rogue device with the A by hijacking a pairing session involving a legitimate device D. To establish a secret key with the A, the adversary must modify the DH public number z_D of D into its own DH public number z'_D, where z_D is contained in the first message m_D sent from D to the A (similar to a typical MitM attack against a DH key exchange).

However, m_D is protected by our integrity verification primitive. Note that in the HELP primitive, only $h(m_D)$ is encoded using MC ON-OFF keying while m_H is being superimposed. The actual value of m_D follows in plaintext. In our proposed modified DH protocol, both $h(m_D)$ and m_D are encoded using HELP. According to Proposition 1, the adversary's success probability in forging m_D in the HELP primitive is δ. When both $h(m_D)$ and m_D are encoded using HELP, we claim that the adversary's success probability in replacing m_D is upper bounded by δ. This is because in the primitive, the adversary can change m_D into any m'_D with probability 1, but his advantage is limited by the probability of changing $h(m_D)$ into $h(m'_D)$, which is δ. In the pairing protocol, the adversary's success probability of changing m_D into m'_D is less or equal to 1. Thus overall, its success probability is less or equal to δ, which is a negligible function of $|\mathbf{s}|$ (number of ON slots injected by helper during $[h(m'_D)]$). Therefore, the adversary will be unable to pair D' with the legitimate A.

D		M		A
Given ID_D, $(\mathbb{G}, q, g)$		Given $ID_{D'}$, $(\mathbb{G}, q, g)$		Given ID_A, $(\mathbb{G}, q, g)$
Pick $X_D \in_U \mathbb{Z}_q$		$X_{D'} \in_U \mathbb{Z}_q$		$X_A \in_U \mathbb{Z}_q$
$z_D \leftarrow g^{X_D}$		$z_{D'} \leftarrow g^{X_{D'}}$		$z_A \leftarrow g^{X_A}$
$m_D \leftarrow ID_D, z_D$		$m'_D \leftarrow ID_{D'}, z_{D'}$		$m_A \leftarrow ID_A, z_A$
(H active)	$\xrightarrow{[h(m_D), m_D]+m_H}$	Cancel and inject	$\xrightarrow{[h(m'_D), m'_D]}$	
(H active)	$\xrightarrow{\mathrm{AE}(\mathbf{s}, K)}$		$\xrightarrow{\mathrm{AE}(\mathbf{s}, K)}$	Verify & Extract $z_{D'}$
			$\xleftarrow{m_A}$	
		$k'_{D',A} \leftarrow (z_A)^{X_{D'}}$		$k'_{D',A} \leftarrow (z_{D'})^{X_A}$
		$X_{A'} \in_U \mathbb{Z}_q$		
		$z_{A'} \rightarrow g^{X_{A'}}$		
		$m_{A'} \rightarrow ID_{D'}, z_{A'}$		
	$\xleftarrow{m_{A'}}$			
		Recover z'_D		
$k_{D,A'} \leftarrow (z_{A'})^{X_D}$		$k'_{D,A'} \leftarrow (z'_D)^{X_{A'}}$		

Figure 3.6: MitM attack against the key-agreement phase of HELP-enabled pairing protocol.

Pairing D with a Rogue Hub

We now examine whether the adversary acting as a rogue A can pair with a legitimate device D. To do so, the adversary can perform a similar MitM attack as in the uplink direction, by replacing the A's DH public parameter z_A with its own number $z_{A'}$. This step of the MitM attack corresponding to the message sent by M to D after the reception of m_D is shown in Figure 3.6.

For this attack to be successful, the adversary must extract the DH public value z_D so that it can compute $k_{D,A'} = (z_D)^{X_{A'}}$. The value of z_D is carried in $[h(m_D), m_D] + m_H$, using the HELP primitive. To recover m_D, the adversary must be able to determine the location vector $\mathbf{s}$ that is used to generate m_H for the portion that corresponds to the transmission of m_D. However, $\mathbf{s}$ is transmitted from H to A using the authenticated encryption function AE$(\cdot)$, so M cannot obtain $\mathbf{s}$ directly from the encrypted version of it.

Alternatively, M can collect and analyze the transmitted signal of $[h(m_D), m_D] + m_H$ after receiving it and attempt to identify all the ON slots in m_H using radio fingerprinting methods [95–100]. However, none of the fingerprinting methods can achieve 100% accuracy. As long as M infers H's ON slots with some probability smaller than one, we can drive the probability of successfully extracting m_D arbitrarily low by increasing the number of slots carrying m_D.

In the following proposition, we derive the probability of D successfully pairing with a rogue A, when the ON slots of the helper are inferred with probability p'_I. Note that in general p'_I is different than the p_I of Proposition 1. The inference of the helper's ON slots in Proposition 1 must occur based on very few samples because the adversary must quickly decide whether to perform signal cancellation. In the rogue A case, the adversary can analyze $[h(m_D), m_D]+m_H$ based on all the samples, so it is expected that $p'_I > p_I$.

Proposition 2. *A legitimate device D pairs with a rogue A with probability $\delta + \epsilon$, where*

$$\delta \;=\; (p'_I)^{|\mathbf{s}'|}, \tag{3.5}$$

and ϵ is a negligible function of the hash length. Here $|\mathbf{s}'| < |\mathbf{s}|$ corresponds to the number of helper's ON slots only during the transmission of m_D in the $[h(m_D), m_D]$, p'_I is the probability of inferring the helper's activity during one MC ON-OFF bit when D and H do not co-transmit, and δ is a negligible function of $|\mathbf{s}'|$ when $p'_I < 1$.

Proof. Assume that the adversary M wants to decode the m_D which contains the key public parameter z_D from $[h(m_D), m_D] + m_H$ without the knowledge of set $\mathbf{s}$.

For $[h(m_D), m_D]$ a bit zero corresponds to (OFF, ON) whereas a bit one corresponds to (ON, OFF). With superimposing H's signal, the A will also receive slots combinations of (ON, ON). The adversary can extract some information of m_D from the (OFF, ON) and (ON, OFF) slots in the $[h(m_D), m_D] + m_H$. But to extract the information from (ON, ON) slots without the knowledge of $\mathbf{s}$. The adversary has to make intelligent guesses for received (ON, ON) slots, which is parameterized as the probability of inferring the helper's activity by M.

Let p'_I be the inference probability for detecting the presence of H's signal. This is discussed in details in Section 3.5. Note that, if H's signal is wrongly inferred (with probability $(1 - p'_I)$), M maps the received bit on which H was active to a wrong outcome.

The adversary makes wrong mapping when it receives (ON, ON) slots on received $[h(m_D), m_D] + m_H$. It happens when M cannot detect the presence of the helper's signal on the slot where D has injected no energy.

$$p_r = \Pr[\text{wrong inference}] = (1 - p'_I). \tag{3.6}$$

In (3.6), p'_I is the probability that M detects the H's signal correctly on a particular bit.

The probability δ of extracting correct m_D from received signal $[h(m_D), m_D]+m_H$ by M. The adversary can decode correct m_D if *none of the bits* are decoded wrong. Each bit is wrongly mapped with probability p_r, given by (3.6). As rejection on each slot occurs independently, the overall probability of correctly decoding m_D from $[h(m_D), m_D] + m_H$ is computed via the Binomial distribution with parameter p_r. That is,

$$\begin{aligned}
\delta &= 1 - \sum_{x=1}^{|\mathbf{s}'|} B\left(x, |\mathbf{s}'|, p_r\right) \\
&= 1 - \sum_{x=0}^{|\mathbf{s}'|} B\left(x, |\mathbf{s}'|, p_r\right) + B\left(0, |\mathbf{s}'| p_r\right) \\
&= \left(1 - p_r\right)^{|\mathbf{s}'|} \\
&= \left(1 - (1 - p'_I)\right)^{|\mathbf{s}'|} \\
&= \left(p'_I\right)^{|\mathbf{s}'|}.
\end{aligned} \tag{3.7}$$

where $B(\alpha, \beta, \gamma)$ is the Binomial probability density function and $|\mathbf{s}'| \subset |\mathbf{s}|$, which corresponds to the number of helper's ON signals only during the transmission of m_D in the $[h(m_D), m_D]$.

We now show that δ is a negligible function of $|\mathbf{s}'|$.

In (3.7), δ is a negligible function if $(1-p_r)^{|\mathbf{s}'|}$ is shown to be a negligible function. To prove the latter, let $\mu(|\mathbf{s}'|) = a^{-|\mathbf{s}'|}$ where $a = \frac{1}{1-p_r}$. For $\mu(|\mathbf{s}'|)$ to be a negligible function, $\forall\ c \in \mathbb{N}$ there exists a $n_0 \in \mathbb{N}$ such that $|\mathbf{s}'| > n_0$ and $\mu(|\mathbf{s}'|) < n^{-c}$. Let $n_0 = c^{\frac{a}{a-1}}$. Then

$$\begin{aligned} a^{|\mathbf{s}'|} &= (a^{\log_a |\mathbf{s}'|})^{-\frac{|\mathbf{s}'|}{\log_a |\mathbf{s}'|}} \\ &= (|\mathbf{s}'|)^{-\frac{|\mathbf{s}'|}{\log_a |\mathbf{s}'|}}, \end{aligned}$$

Since $|\mathbf{s}'| > n_0$, it follows that

$$\begin{aligned} \frac{|\mathbf{s}'|}{\log_a |\mathbf{s}'|} &> \frac{n_0}{\log_a n_0} \\ &> \frac{n_0}{n_0^{\frac{1}{a}}} \\ &> c. \end{aligned}$$

Therefore,

$$\begin{aligned} \mu(|\mathbf{s}'|) &= a^{-|\mathbf{s}'|} \\ &= (|\mathbf{s}'|)^{-\frac{|\mathbf{s}'|}{\log_a |\mathbf{s}'|}} \\ &< n^{-c}. \end{aligned}$$

This proves that $(1-p_r)^{|\mathbf{s}'|}$ is a negligible function for $a \neq 1$ or equivalently $p_r \neq 0$.

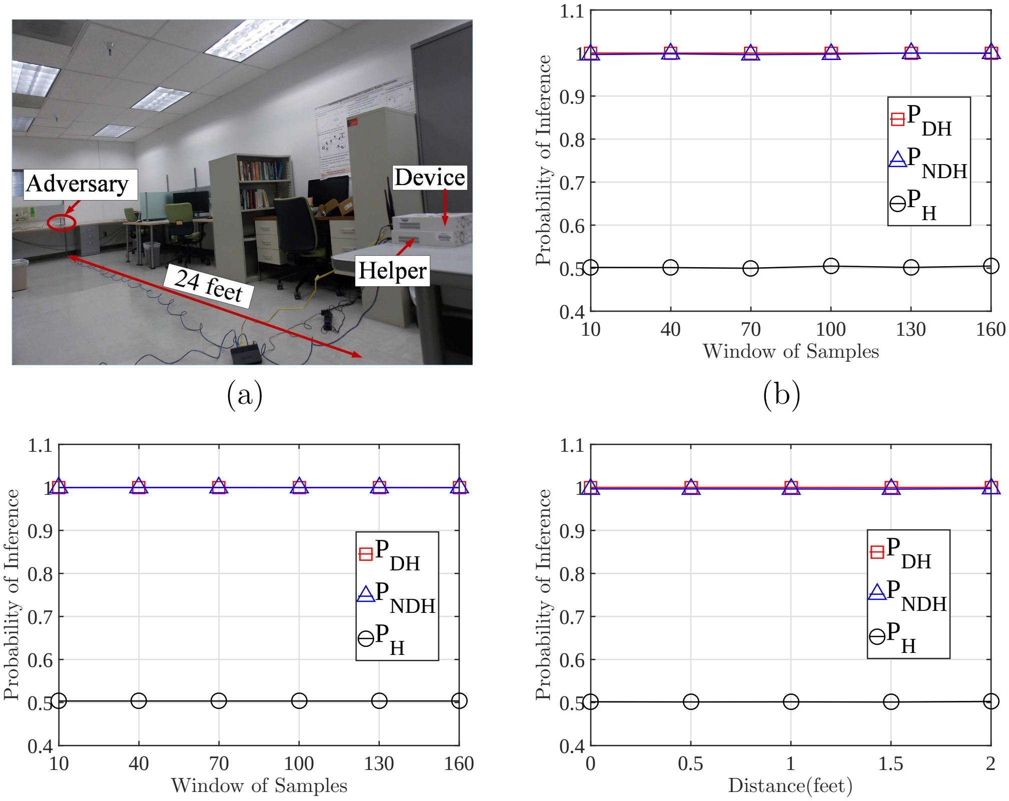
After the attacker extracts m_D, the rogue A needs to pass the challenge-response authentication in the key confirmation phase. Assuming the use of a strongly universal hash function to compute the response $h_{k_{D,A'}}(ID_A||C_D||0)$, he can only pass this authentication if he has the correct key $k_{D,A'}$. Otherwise, his successful probability ϵ is negligible. But he can only obtain the correct key by extracting the correct m_D value. Therefore, the success probability of the rogue A to pair with the device is upper bounded by $\delta + \epsilon$, where ϵ is a negligible function (of the length of the hash function). Since δ is a negligible function of $|\mathbf{s}'|$ which can be the same as the message length (and here the m_D is a DH public number, whose bit length is typically larger or equal to the hash length), the overall probability is a negligible function. This concludes the proof.

3.5 Evaluation

3.5.1 Helper Activity Inference

In this section, we first analyze M's capability in timely identifying the helper's ON slot when the helper is transmitting the ON-OFF message m_H. For this purpose, the adversary could employ several PHY-layer characteristics of the helper's transmission to pinpoint when H is active. These include (a) the received signal strength [95], (b) the frequency offset [96], (c) the channel impulse response $\mathbf{h}_{H,A}$ [97], (d) the I/Q origin offset [98], (e) the transient radio state [99], and (f) the angle of arrival for the incoming signal [100].

We first examine M's attempt to perform the signal cancellation and injection required by the MitM attack of Figure 3.6. To avoid rejection of m'_D by the A, the adversary has to *swiftly detect* a helper's ON slot and decide whether to perform signal cancellation. Most existing radio fingerprinting methods are not suitable for such quick online detection. The frequency offset and channel impulse response are estimated using known preambles that are typically included in headers. Such preambles do not precede the helper's ON slots. The I/Q origin offset is not a suitable method because we employ ON-OFF modulation for message transmission. The methods that detect the transient state of a radio when it turns on can only be used to identify the start of a transmission (although an ON-OFF modulation implies a transition from an OFF to an ON state, the radio transmitter is powered through the entire transmission of an ON-OFF signal and a transient state is not observed with every slot). Differentiating between D and H using an AOA requires a very narrow directional beam due to the proximity between H and D. Such



(a) (b)

(c) (d)

Figure 3.7: (a) Experimental setup, (b) detection probability as a function of the window of samples when the power at H and D is fixed, (c) detection probability as a function of the window of samples when the power at H and D varies, and (d) detection probability as a function of the distance between D and H, when H and D remain equidistant from M.

narrow beamwidths can be achieved by using an antenna array [104] or a parabolic antenna [105]. However, the hardware cost is prohibitive and the antenna would be quite visible. For example, an adversary at 50ft from D and H requires two 50-element antenna arrays pointed to D and H respectively via the LoS path, to differentiate between D and H when their distance is set to 4ft. This calculation assumes a 2.4GHz operating frequency.

Fast Helper Detection based on RSS

The simplest and most timely method for detecting the presence of the helper is to measure the received signal strength over some small number of samples at the beginning of every slot. Let b_D and b_H represent the bit simultaneously transmitted by D and H respectively over two slots t_i and t_{i+1}. There are four possible bit combinations that yield two candidate power profiles for $b_D + b_H$, as measured by the adversary. When $b_D = b_H$, the helper and D overlap in one of the two slots (either t_i or t_{i+1}), depending on the value of b_D, b_H. In this case, one of the slots is OFF whereas the other slot is ON with a significantly higher power because the two ON slots of H and D are superimposed (here, we have considered the worst-case scenario and ignored the possibility of destructive interference). We expect that M will be able to infer the ON slot of the helper with probability $p_I = 1$, due to the higher RSS value of the first few samples of the ON slot.

When $b_D \neq b_H$, both t_i and t_{i+1} are ON and have similar power profiles if H and D transmit with the same power and are placed in close proximity. In this case, the adversary is expected to be unable to differentiate a helper's ON slot from a device's ON slot with the probability much higher than a random guess. The four possible cases for one slot observed by the adversary are: (a) P_1 : both H and D are ON, (b) P_2 : H is ON and D is OFF, (c) P_3 : D is ON and H is OFF, and (d) P_4 : both H and D are OFF. For each case, the adversary determines four threshold values $E[P_1], E[P_2], E[P_3]$, and $E[P_4]$, that represent the average expected power, as measured by the first few samples of a slot.

Without loss of generality, let $E[P_1] > E[P_2] > E[P_3] > E[P_4]$.[2] Let also $E[P(t_i)]$ denote the average power measured over slot t_i using the first few samples. The adversary classifies t_i to one of four cases by mapping $E[P(t_i)]$ to the closest threshold. That is, case P_1 is inferred if $E[P(t_i)] > \frac{E[P_1]+E[P_2]}{2}$, case P_2 is inferred if $\frac{E[P_1]+E[P_2]}{2} \leq E[P(t_i)] < \frac{E[P_2]+E[P_3]}{2}$, etc. A wrong inference is made when $E[P(t_i)]$

[2] $E[P_2]$ and $E[P_3]$ can be similar but not exactly the same, so we can assume some ordering to make a classification rule.

that belongs to case P_i is mapped to a case P_j with $P_i \neq P_j$. In Proposition 1, we have assumed that the probability p_I for correctly inferring cases P_1 and P_4 is equal to one. In P_1, the RSS is expected to be relatively high due to the co-transmission from D and H. In P_4, the RSS is expected to be low because neither D nor H are transmitting. However, the thresholds for cases P_2 and P_3 are expected to be very close, thus leading to frequent wrong inferences. We experimentally verify this claim.

Experimental Evaluation of p_I:

Experimental setup: To evaluate p_I, we setup three NI-USRP 2921 devices in an indoor laboratory environment. Two USRP devices represented D and H, whereas a third USRP device is placed at 24 feet away acting as an adversary. The transmit power for an ON slot was set to 20dBm for both D and H with a symbol duration of 1ms. The devices were set to work at 2.4GHz and were synchronized. The sampling frequency was set to 2MHz. We tested two scenarios: (1) H is stacked on top of D, and (2) H is moved away from the legitimate device. The experiment setup is shown in Figure 3.7(a).

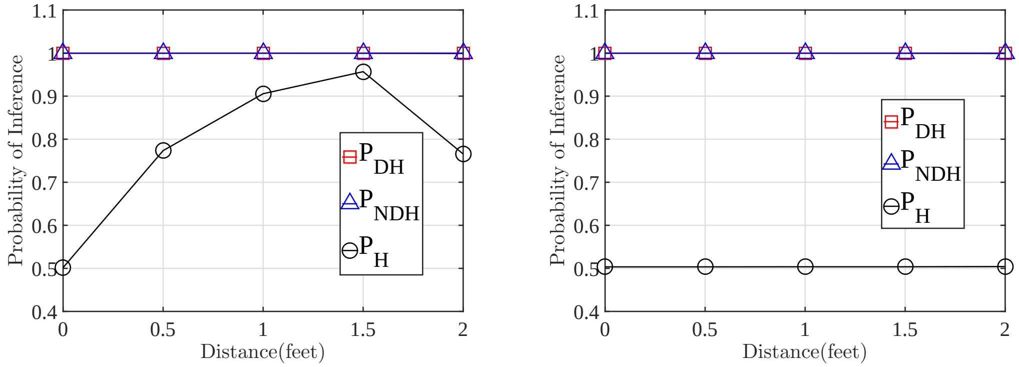
We implemented amplitude shift keying (ASK) to transmit MC ON-OFF coded messages and repeatedly transmitted message $\{1, 0, 1, 0\}$ from D and message $\{1, 1, 0, 0\}$ from H simultaneously. The signals from H are MC-coded only when the bit value is one. The superposition of the two signals implemented all four cases P_1-P_4.

Results: Let P_{DH} denote the probability of detecting that D and H transmit simultaneously, P_{NDH} denote the probability of detecting that neither D nor H transmit, and P_H denote the probability of detecting that H is transmitting alone. These correspond to p_I for any of the candidate scenarios. In the first experiment, we measured the detection probability as a function of the sampling window size used for computing the average RSS value for a given slot. Intuitively, a longer sampling window would lead to better inference but will delay the cancellation operation. Figure 3.7(b) shows the resulting detection probabilities as a function of the sample window. We observe that the detection probabilities P_{DH} and P_{NDH} are relatively low and are further reduced with the increase of the sample window. However, the detection probability P_H is close to 0.5 irrespective of the sample window size. This indicates that differentiating between the ON slots of the helper and of the legitimate device, when only one of the two transmits, is practically equivalent to a random guess. Our results justify the selection of $p_I = 1$ when the H and D are simultaneously absent or present, and $p_I = 0.5$ otherwise.

In the second experiment, we repeated the first experiments but configured H and D to vary their transmission power on a per-slot basis. The power was varied to reduce the inference capability of M. Specifically, H and D oscillated their power at random between 10dBm and 20dBm. Figure 3.7(c) shows the detection probabilities as a function of the window of samples used for inference.

Effect of proximity on p_I: We further performed experiments to evaluate the effect of the proximity between D and H on their distinguishability. We repeated the first experiment and varied the distance between H and D. In the first part of the experiment, H was moved away from D while keeping the D-M and H-M distances similar (the helper's motion was perpendicular to the D-M line. Figure 3.7(d) shows that the detection probability for each case is similar to the case where H is stacked on top of D. In the second part of the experiment, H was moved towards M, and therefore, the distance between H and M was gradually reduced. Figure 3.8(a) shows the respective detection probabilities. As expected, decreasing the distance between M and H improves the adversary's inference capability, but the inference remains imperfect when D and H remain relatively close.

In the fourth experiment, we repeated the second part of the third experiment but configured H and D to vary their transmission power on a per-slot basis. The power was varied to reduce the inference capability of M. Specifically, H and D oscillated their power at random between 10dBm and 20dBm. Figure 3.8(b) shows the same results when the distance between D and H was also varied, with H moving towards M. We observe that P_H remains a random guess even when H is



(a)

(b)

Figure 3.8: (a) Detection probability as a function of the distance between D and H when H is moved towards M, and (b) detection probability as a function of the distance between D and H when H is moved towards M, when D and H are transmitting random powers.

moved away from D (comparison of P_H in Figures 3.8(a) and 3.8(b)), indicating that a power variation approach can account for situations where H is not placed exactly on top of D. Distinguishing signals from D and H using RSS remains a random guess even when H is 2ft away from D.

Fast Helper Detection Based on Time

In this section, we discuss an inference technique that exploits the possible time misalignment between the transmissions of H and D due to clock drift and different path delays to the receiver. There have been extensive studies on synchronization of independent wireless nodes, but practically it is impossible to reach perfect synchronization [90]. The adversary can exploit the synchronization offset between H and D to infer the presence of helper's ON signals. If H is faster (slower) than D, the ON slots of H will appear slightly earlier (later) than the ON slots of D. An example of a fast H is shown in Figure 3.9, where there is a synchronization offset ϵ between D and H. If M fixes his clock to H, it can infer the presence of helper's ON slots without having to resort to RSS estimation. It should be noted here, the

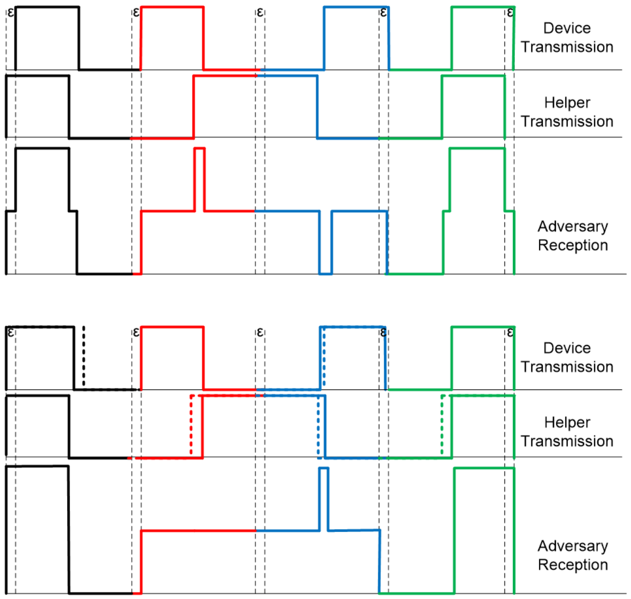


Figure 3.9: Synchronization offset without and with randomized start time of each bit.

A performs detection of ON slots by taking an average value of the power of all the samples. Therefore, a perfect synchronization between D and H is not required for the correctness of the proposed protocol.

To prevent the inference of the helper's ON slots based on time misalignment, we randomize the start times of each bit (first slot of the MC ON-OFF bit) both at H and D. Specifically, a random time offset ϵ, positive or negative, is selected from a uniform distribution $\mathcal{U}(\epsilon_l, \epsilon_h)$. The lower bound ϵ_l is selected to be the maximum synchronization error between D and H. This can be calculated as the expected clock drift over the transmission time of H plus a maximum time difference between path delays. The upper bound τ_h can be some reasonable value (e.g., $2\epsilon_l$). Moreover $\tau << t$, where t is the slot duration. This will ensure the correct sequence decoding at the A. The lower part of Figure 3.9 shows an example of applying the randomized start time for each bit. We observe that no device is always faster (slower), thus preventing M from fixing its clock to H.

Experimental Evaluation of p_I: To verify the validity of our time random-

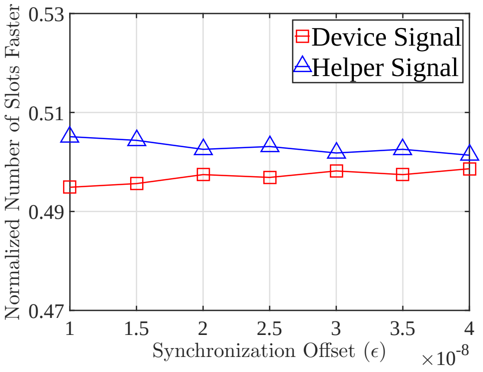
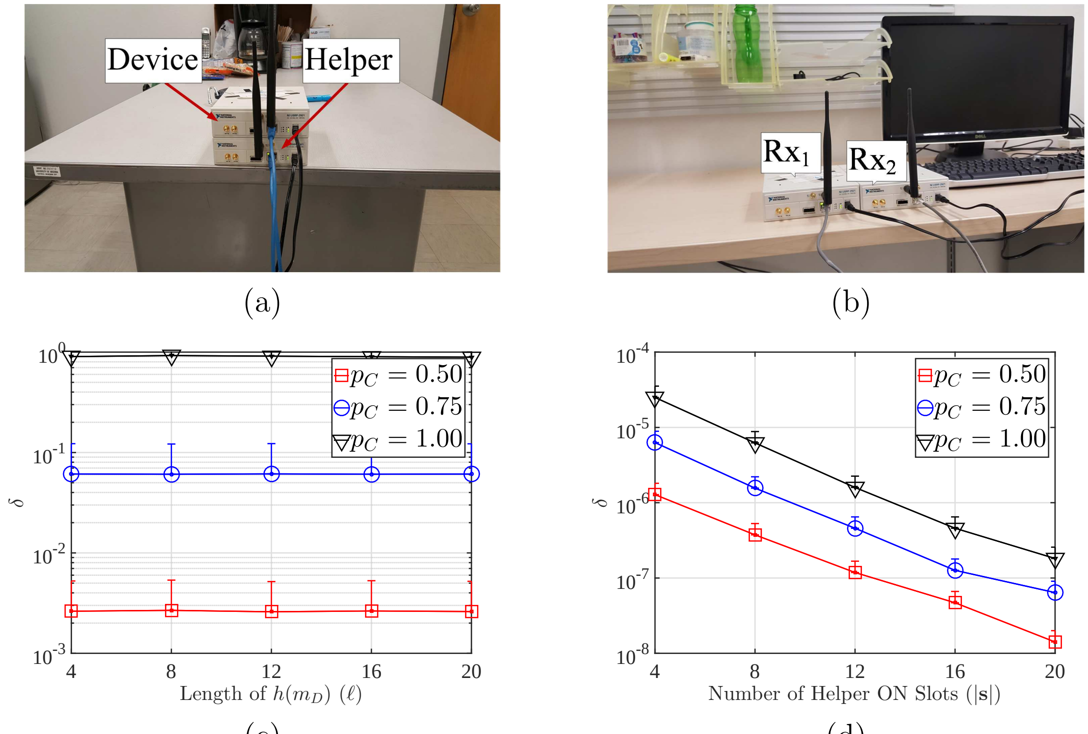


Figure 3.10: Fraction of slots that one device is faster than the other as a function of the delay offset ϵ.

ization approach and its impact on the inference probability p_I, we setup three NI-USRP 2921 devices in an indoor laboratory environment as D, H, and M, respectively. As in previous experiments, H was stacked on top of D, whereas M was placed 24 feet away from D, H. The transmit power for an ON slot was set to 20dBm with a symbol duration of 1ms. An artificial clock misalignment $\tau = 0.1$msec was set between H and D to emulate the maximum synchronization error. We then varied the random time offset ϵ selected by H and D. The experiment lasted for the transmission of 10^6 sequences of 40 bits each.

Figure 3.10 shows the fraction of slots for which each device was detected to be faster as a function of the maximum synchronization error ϵ. We observe that for sufficiently high values of ϵ, H is almost 50% of the time faster than D. Practically, using time misalignment to distinguish the helper becomes a random guess. □

In Proposition 2, δ depends on two variables; the cardinality of set $\mathbf{s}'$ which is a subset of $\mathbf{s}$ corresponding to H's ON signal only during the transmission of m_D in $[h(m_D), m_D]$, and the inference probability of the helper's activity during the transmission of $[h(m_D), m_D] + m_H$, which is p'_I. From eq. (3.5), it is evident that δ is a negligible function of $|m_D|$, and a monotonically increasing function of p'_I. In Figure 3.12, we show δ as a function of $|\mathbf{s}'|$ for various values of p'_I and fixed



(a)

(b)

(c)

(d)

Figure 3.11: (a) Placement of D and H, (b) placement of the A (RX_1) and RX_2. (c) probability of acceptance of a modified message at the A in the absence of H, and (d) probability of acceptance of a modified message at the A in the presence of H.

hash length of $\ell = 160$. As expected, a higher p'_I yields a higher δ value for the adversary. For instance, when $p'_I = 0.9$, $\delta = 0.0018$, when $|\mathbf{s}'| = 80$, which may not be acceptable. However, doubling the size of $\mathbf{s}'$ lowers δ to 5×10^{-8}. Note that, such an attack has to happen in an online manner. This is because the rogue A must pass the challenge-response phase from the device in the key confirmation phase, so the attacker only has one chance to guess $\mathbf{s}$ and derive a probable DH key from the guessed z_D, which is only successful with small probability δ (similar to limited-guess online password attacks).

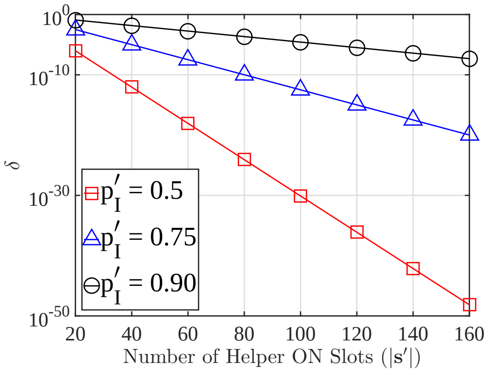


Figure 3.12: Probability of pairing with a rogue A as a function of $|\mathbf{s}|$, for varying inference capabilities of helper activity.

3.6 Chapter Summary

We considered the problem of pairing two devices using in-band communications in the absence of prior shared secrets. We proposed a new PHY-layer integrity protection scheme called HELP that is resistant to signal cancellation attacks. Our scheme operates with the assistance of a helper device that has an authenticated channel to the A. The helper is placed in close proximity to the legitimate device and simultaneously transmits at random times to allow the detection of cancellation attacks at the A. We showed that a pairing protocol such as the DH key agreement protocol using HELP as an integrity protection primitive can resist MitM attacks without requiring an authenticated channel between D and the A. This was not previously feasible by any of the pairing methods if signal cancellation is possible. We studied various implementation details of HELP and analyzed its security. Our protocol is aimed at alleviating the device pairing problem for IoT devices that may not have the appropriate interfaces for entering or pre-loading cryptographic primitives.

CHAPTER 4

IN-BAND SECRET-FREE PAIRING PROTOCOL FOR COTS WIRELESS DEVICES

4.1 Introduction

4.1.1 Motivation

HELP as well as most in-band pairing protocols [31–35], rely on Manchester coded ON/OFF keying to thwart active attacks. The adoption of those methods may require firmware/hardware modifications. To be compatible with Commercial-Off-The-Shelf (COTS) devices, we proposed a novel message integrity and authentication primitive that does not rely on any special modulation. In this chapter, we study the problem of *secure in-band pairing for devices that do not share any prior secrets*. We develop a secret-free in-band trust establishment primitive, called SFIRE for short, that draws security from hard-to-forge signal propagation laws. The primary operational scenario for SFIRE is shown in Figure 4.1. A user executes a pairing session between the legitimate device D and the hub A. During pairing, M launches an MitM attack over the wireless channel to establish a key with the hub and/or the device. In SFIRE, active attacks are detected by correlating RSS fluctuations measured simultaneously at A and a helper device H, while the pairing device is active. RSS has been explored in several prior works for device authentication [72, 76] however, these methods require firmware and/or hardware alterations.

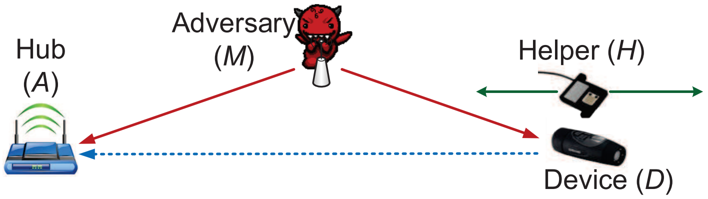


Figure 4.1: The basic system model depicting all the entities.

4.1.2 Main Contributions and Chapter Organization

Our contributions: Our main contributions are four-fold:

- We develop a novel PHY-layer primitive called SFIRE that prevents rogue devices from joining the network. SFIRE is resistant to an MitM attacker, capable of advanced signal manipulations. SFIRE's security relies on a novel "RSS authenticator" that exploits physical signal propagation laws to thwart attackers.

- We use SFIRE to construct a secure in-band pairing protocol based on the Diffie-Hellman (DH) key agreement [49]. Our protocol allows a legitimate device join a hub and establish a pairwise key. One notable feature of our protocol is that it does not require any hardware/firmware modifications or special transmission modes for the device. This makes SFIRE interoperable with any commercial off-the-shelf (COTS) device that has a common wireless link with the hub.

- We theoretically explore the security of the RSS authenticator under worst-case scenarios. We analyze the ability of active adversaries with increasing capabilities (antenna directionality, transmission power control, etc.) to defeat SFIRE.

- We carry out extensive experimentation to establish the distinct RSS features that can be used for message integrity verification. We analyze the security of SFIRE under active adversaries with increasing capabilities. We implement SFIRE on COTS equipment and USRPs to validate the offered security. Our experiments attest the theoretical findings and verify the resistance to active signal manipulations, even if the adversary enjoys favorable channel conditions to the hub and the helper.

Chapter Organization: The remainder of this chapter is organized as follows. In Section 4.2, we describe the system and adversary models. We present the SFIRE primitive and secure pairing protocol in Section 4.3. We theoretically and experimentally evaluate the PHY-layer features exploited in "RSS authenticators" of SFIRE in Section 4.4 and Section 4.5 respectively. We evaluate the security of pairing protocol and various implementation details in Section 4.6 and conclude in Section 4.7.

4.2 Model Assumptions

4.2.1 System Model

The following entities are part of the system model.

Hub (A): The hub coordinates the secure pairing process. It is responsible for the authentication of the legitimate device and the coordination with the helper device.

Legitimate Device (D): A COTS device which attempts to pair with A in-band. Pairing results in the establishment of a secret key. D does not share secrets with A before pairing. It is assumed to be under the user's control.

Helper Device (H): The helper is a trusted device such as a smartphone that is under the user's control. It assists A with the pairing process and already shares a secure authenticated channel with A. However, H does not share any secrets with D. This channel is established via conventional means such as loading a common key. Using this secure channel, H can apply an authenticated encryption function

AE($\cdot$) on any transmission to guarantee the message confidentiality and integrity, and the authenticity of the source. Any such AE($\cdot$) can be utilized with the proposed protocol. For example, if H and A share a public/private key pair, H can encrypt–sign–encrypt (or sign–encrypt–sign), or if they share a common master symmetric key, an encrypt-then-MAC operation can be followed to implement AE($\cdot$), after separate symmetric keys are generated from the master key for the encryption and MAC operations. We refer the reader to [93] for more details on authenticated encryption. Note that pairing H to A is a one-time effort and is not repeated with every device join. We believe that this is an acceptable tradeoff for pairing many COTS devices. Finally, H is assumed to be loosely synchronized to A using any known method (e.g., [90]).

4.2.2 Threat Model

Adversary (M)**:** We consider an active adversary that controls one or more adversarial devices. We assume that M cannot get very close to the helper and the legitimate device (e.g., within 1-2 meters) as it will become noticed by the user. M's goal is to either pair with A as a legitimate device or spoof a rogue hub that pairs with D. Because device pairing is initiated by the user, M attempts to realize his goal by launching a MitM attack during a pairing session. The MitM attack is performed by canceling/overshadowing signals at D, A and H and injecting rogue messages. The adversary is aware of the protocol executed by the legitimate devices, but does not have physical access to any of them. Denial-of-service (DoS) attacks such as jamming, are orthogonal to our studies. Moreover, as commonly assumed, M is incapable of physically blocking signals (*e.g.*, by adding a Faraday cage) around D, A, or H. We consider three adversary types with increasing capabilities.

Type 1: A type 1 adversary can perform an overshadowing attack [54] to inject his own message at H and A using omnidirectional transmsissions.

Type 2: A type 2 adversary is a type 1 adversary that additionally employs coordinating devices with directional antennas that can target individual devices.

Type 3: A type 3 adversary is a type 2 adversary that additionally applies fine-grained power control to achieve any desired RSS profile.

4.3 The SFIRE Protocol

SFIRE is an in-band pairing protocol that does not require secret preloading. Authentication is achieved via a novel PHY-layer protection primitive which we call as an "RSS authenticator". We first describe the RSS authenticator and then use it to construct SFIRE.

4.3.1 Constructing an RSS Authenticator

Referring to the basic scenario of Figure **??**, consider D attempting to pair with A. Let D transmit m_D in plaintext because D and A do not share any prior security association. While m_D is transmitted, H is swept over D in an oscillating motion, with both H and A simultaneously measuring the RSS. H relays the received message, say m'_D and the associated RSS samples to A via their shared authenticated channel. The hub compares m'_D with its own received message m''_D and also computes the RSS ratio between the samples sent from H and its own samples. The hub uses the RSS ratio fluctuation patterns to verify that m''_D indeed originated from D. Formally, the authentication steps are as follows.

1. **Initialization:** The user presses a button on D or simply switches D on to set it to pairing mode. The user then presses a button or a virtual button on H to initiate the protocol. H sends an authenticated *request-to-communicate* message to A using the AE($\cdot$) function, which attests that D is present. The hub starts a timer.

2. **Transmission of m_D:** D broadcasts m_D a total of k times in plaintext using back-to-back frames. The repetition of m_D bridges the time scales between

message transmission and the user actions, as the latter are several orders of magnitude slower.

3. **Sweeping motions:** While m_D is transmitted, the user sweeps H over D (see Figure 4.3(a)). A sweeping motion is defined as a continuous motion passing over D. While in motion, H decodes messages $m'_D(1), \ldots, m'_D(k)$ and samples the RSS at a fixed rate. Let $\mathbf{r}_H = \{r_H(1), \ldots, r_H(n)\}$ by the RSS sequence with $t_H(1)$ denoting the timestamp of the first sample.

4. **Reception of m_D at A:** The hub decodes $m''_D(1), \ldots, m''_D(k)$. The hub also records $\mathbf{r}_A = \{r_A(1), \ldots, r_A(n)\}$, and the reception time $t_A(1)$ of the first sample.

5. **Authentication at H:** The helper checks if $m'_D(1) \overset{?}{=} \cdots \overset{?}{=} m'_D(k)$. If not, H sends an AE(ABORT) message to A via their shared authenticated channel. If the decoded messages match, H compiles message $m_H = \{\mathbf{r}_H, m'_D(1), t_H(1)\}$ and sends AE(m_H) to A.

6. **Authentication of m_H:** The hub decrypts m_H and verifies its integrity using VD($\cdot$), which is the corresponding authentication/integrity verification function to AE($\cdot$). If verification fails, A aborts m''_D.

7. **Authentication of m_D:** The hub first verifies that $m''_D(1) \overset{?}{=} \cdots \overset{?}{=} m''_D(k)$. If verification fails, it aborts the pairing process. If successful, the hub verifies $m''_D(1) \overset{?}{=} m'_D(1)$. If verification fails, the hub aborts the pairing process. Otherwise, A proceeds to the RSS authentication. The hub uses the timestamps $t_H(1)$ and $t_A(1)$ to align $\mathbf{r}_H$ with $\mathbf{r}_A$. The hub computes the RSS ratio (Γ) between $\mathbf{r}_H$ and $\mathbf{r}_A$:

$$\Gamma = \{\gamma(1), \gamma(2), \ldots, \gamma(n)\},\ \gamma(i) = \frac{r_H(i)}{r_A(i)}.$$

The hub performs a set of RSS authentication tests to verify the authenticity of m''_D. If any of the tests fails, the pairing is aborted and the user has to

D		A
Given ID_D,		Given ID_A,
$(\mathbb{G}, q, g)$		$(\mathbb{G}, q, g)$
Pick $X_D \in_U \mathbb{Z}_q$		$X_A \in_U \mathbb{Z}_q$
$z_D \leftarrow g^{X_D}$		$z_A \leftarrow g^{X_A}$
$m_D \leftarrow ID_D, z_D$	$\xrightarrow{[m_D]}$	$m_A \leftarrow ID_A, z_A$
(H active)	$\xrightarrow{\text{AE}(m_{H,j},K)}$	Verify
		& Extract z_D
Verify & Extract	$\xleftarrow{\text{AE}(m_A,K)}$	
m_A at H		
	$\xleftarrow{[m_A]}$	(H active)
$k_{D,A} \leftarrow (z_A)^{X_D}$		$k_{D,A} \leftarrow (z_D)^{X_A}$

Figure 4.2: DH key agreement using SFIRE as a message authenticator.

restart the pairing process. If all test pass H displays SUCCESS. If a timer at A expires, the pairing process fails.

4.3.2 SFIRE-enabled Device Pairing

Parties A and D can securely establish a pairwise key by integrating SFIRE to the DH key-agreement protocol. The SFIRE-enabled DH message exchange is shown in Figure 3.3. The hub (or D) use public parameters $(\mathbb{G}, q, g)$ of the DH scheme, where ($\mathbb{G}$ is a cyclic group of order q and g is a generator of G). Device D computes $z_D = g^{X_D}$, where X_D is chosen from $\mathbb{Z}_q$ uniformly at random. After the initialization step (omitted from Figure 3.3), D broadcasts $m_D : ID_D, z_D$ in plaintext to A. The hub verifies this broadcast using SFIRE. In the protocol of Figure 3.3, messages protected by SFIRE are denoted by $[\cdot]$. The hub replies with $z_A = g^{X_A}$, where X_A is chosen in $\mathbb{Z}_q$ uniformly at random. Each party independently computes $k_{D,A} = g^{X_D \cdot X_A}$. Immediately following the key-agreement, D and A engage in a key confirmation phase, initiated by D. This can be done by executing a two-way challenge-response protocol [102]. If any of the parties fails verification, it sends an abort message.

4.3.3 Securing the Downlink Direction

In the DH exchange of Figure 3.3, the authenticity of m_A is not verified at D. A MitM adversary acting as a rogue hub may attempt to pair with D by replacing m_A with its own message. However, this will result in an incomplete session at A. In this case, A can notify H of the incomplete pairing that displays a failure message. The user can then re-initiate the pairing protocol.

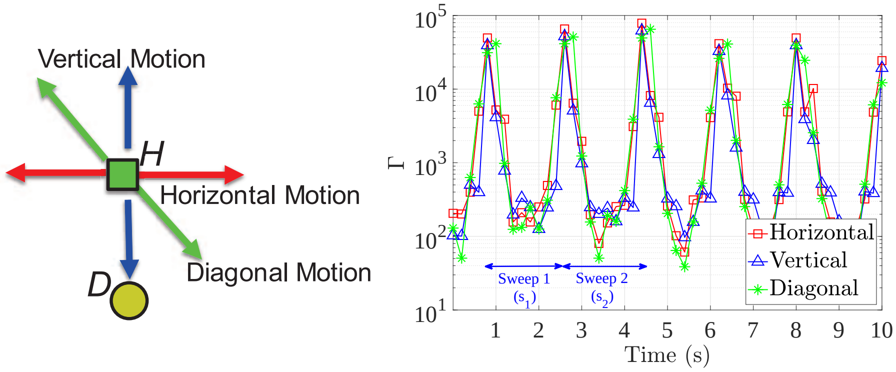
Message m_A can be explicitly authenticated by increasing human effort. After verifying and accepting m_D, A transmits m_A to H using AE($\cdot$). Then A sends m_A in plaintext to D. Device D records m'_A and the corresponding RSS values as dictated in step 4 of the SFIRE protocol. The helper repeats the transmission of m_A while it is being swiped over D several times. The device decodes m''_A and records the RSS values. To deem m_A authentic, it must hold that $m'_A \overset{?}{=} m''_A$ and the first three RSS authentication tests are passed at D. Note that the helper does not relay any RSS measurements to D, but D directly measures RSS from the respective transmissions of H and A. D does not need special hardware, as RSS measurements are readily available in-band.

4.4 RSS Authentication Tests

We now describe four RSS authentication tests performed by A to verify m_D. Tests are introduced to mitigate adversaries with increasing capabilities. Three of our tests rely on identifying the samples that belong to each sweep performed by the user. The hub organizes the RSS samples Γ in sweeps as follows:

Definition 1. *Sweep* $\mathbf{s}_i$: Let Γ be a set of RSS ratio samples ordered according to time. Let $\mathcal{F}$ be the fitted smooth curve on Γ. A sweep $\mathbf{s}_i$ is a set of samples $\{s(i,1), s(i,2), \ldots, s(i,w_i)\}$, where $s(i,1)$ and $s(i,w_i)$ are the samples closest to the i^{th} and $i+1^{st}$ local maximum of $\mathcal{F}$, respectively.

We use a fitted smoothed curve [106] to address the temporal RSS ratio variation.



(a) (b)

Figure 4.3: (a) Various sweeping motions of H over D, (b) RSS ratio fluctuation as a function of time for various motions.

Although the RSS ratio is expected to be proportional to distance (especially in the presence of a strong LoS component), the RSS would vary with nearby movement. The fitted smooth curve allows us to uniquely define local maxima. When a point from Γ is assigned to a sweep $\mathbf{s}_i$, it is removed from Γ such that sweeps form disjoint sets. If the user does not initiate the device movement close to D where the peak ratio is achieved, the first few samples are discarded until a peak is found. One RSS ratio timeline indicating a sample set Γ based on our experiments (see Section **??** for the experimental setup description) for the three motion types of Figure 4.3(a) is shown in Figure 4.3(b).

4.4.1 Test 1: Peak RSS Ratio

In the first test, the hub compares the largest sample value in every sweep $\mathbf{s}_i$ with a threshold τ_{peak}. The verification passes if there is at least one sample in every $\mathbf{s}_i$ with a value greater than τ_{peak}. This test exploits the short distance between H and D during each sweep and the physical signal propagation laws. When the helper is swept over D, he reaches within a few wavelengths from D, whereas A is expected to be at a significantly longer distance. Due to the proximity of H and D, the peak RSS at H can be several orders of magnitude higher than the RSS at A. The peak

RSS ratio from a remote location M relative to D cannot achieve very high values due to geometric constraints (the distance difference between the M-H and M-A paths becomes smaller as M's location becomes more remote, unless the three are co-linear). Formally, the steps of Test 1 are as follows:

1. **Compile sweeps:** Compile the sweep set $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_\ell$ from sample set Γ according to Definition 1.

2. **RSS ratio test:** If

$$\max_{\mathbf{s}_i}(s(i,j)) \geq \tau_{peak}, \ \forall \ i = 1, 2, \ldots, \ell$$

then D passes Test 1.

Determining τ_{peak}: We now show how to select the threshold τ_{peak} for Test 1. Consider a transmission from D received by H and A simultaneously. Due to the proximity between D and H during the sweeping motion, the D-to-H channel has a strong LoS component. For this topology, the propagation loss can be modeled after the free-space channel model with a path-loss exponent $\alpha_H = 2$ [91]. The D-to-A channel, on the other hand, could adhere to different models depending on the setting. *Given that no single propagation model can capture all scenarios, we consider a general pathloss model where signal attenuation is primarily captured via the attenuation factor α_A that can range from two to five [91].* Under this general model, the RSS ratio γ at A when D transmits is given by:

$$\gamma = \frac{r_H}{r_A} = \frac{G_H}{G_A} \cdot \frac{(d_{DA})^{\alpha_A}}{(d_{DH})^2}, \tag{4.1}$$

where d_{DX} is the distance between D and X, G_X is the antenna gain of X, and α_A is the pathloss factor for the D-to-A channel. To ease our theoretical analysis, we assume that the path loss exponents remain constant during the brief sweeping process and focus on a single sweep. We simplify our notation by dropping the sweep index and focus on sweep $\mathbf{s}$ with samples $\{s(1), s(2), \ldots, s(w)\}$. The maximum RSS ratio in a sweep $\mathbf{s}$ is given by:

$$\begin{aligned}\max_{\mathbf{s}}\left(s(j)\right) &= \max_{\mathbf{r}_H}\left(\frac{r_H(j)}{r_A}\right)\\ &= \max_{\mathbf{d}_{DH}}\left(\frac{G_H\cdot(d_{DA})^{\alpha_A}}{G_A\cdot(d_{DH}(j))^2}\right)\\ &= \frac{G_H\cdot(d_{DA})^{\alpha_A}}{G_A\cdot\min\limits_{\mathbf{d}_{DH}}(d_{DH}(j))^2)}\\ &= \frac{G_H\cdot(d_{DA})^{\alpha_A}}{G_A\cdot(d_{DH}^{\min})^2},\end{aligned} \tag{4.2}$$

where $\mathbf{r}_H$ is the vector of all sampled RSS values at H during sweep $\mathbf{s}$, $\mathbf{d}_{DH}$ is the vector of the the corresponding distances between D and H when D's signal is sampled during the sweep $\mathbf{s}$, and $d_{DH}^{\min}$ is the minimum distance between D and H during the sweep $\mathbf{s}$ (recall that only H is moving during a sweep). When the legitimate device D is transmitting, at least one value in $\mathbf{s}$ must be greater than τ_{peak}. The benign case determines the threshold τ_{peak} that must be used for detecting a Test 1 violation.

$$\tau_{peak} \leq \max_{\mathbf{s}}\left(s(j)\right) = \frac{G_H\cdot d_{DA}^{\alpha_A}}{G_A\cdot(d_{DH}^{\min})^2}. \tag{4.3}$$

The value of τ_{peak} is further used to analyze the security of Test 1. Here, we observe that the τ_{peak} depends on the distances between the legitimate entities and the user's movement. The threshold value is calibrated for legitimate entities and the user is given instructions on the suggested movement patterns (e.g., sweep the helper across D and pass within a few centimeters from D).

One challenge with determining τ_{peak} is that the pathloss α_A between D and A depends on the setting. To make our method applicable to any setting, we consider the worst-case scenario where $\alpha_A = 2.$, i.e., the D-to-A channel is a LoS channel. It is fairly straightforward to observe that τ_{peak} is a monotonically increasing function of α_A. Therefore, τ_{peak} is minimized for $\alpha_A = 2$. Even if the D-to-A channel conforms to a model with higher α_A, a legitimate device will pass Test 1, when τ_{peak} is selected with $\alpha_A = 2$. Finally, selecting τ_{peak} with $\alpha_A = 2$ is the best case for the adversary.

Security Analysis: In an attempt to defeat Test 1, the signal injected by M during sweep $\mathbf{s}_M$ has to achieve a maximum RSS ratio that exceeds τ_{peak}. As discussed earlier, the best case for the adversary is when τ_{peak} is minimized, which is achieved when $\alpha_A = 2$. We analyze the security of Test 1 when

$$\tau_{peak} = {}^{G_H \cdot (d_{DA})^2}\!/\!_{G_A \cdot (d_{DH}^{\min})^2}.$$

Type 1 adversary: To succeed in pairing with the hub, a Type 1 adversary launches an overshadowing attack [54] by transmitting at a desired power using an omnidirectional antenna. Let M attempt to replace D's message m_D with m_M at A from a location L_M. To succeed in injecting m_M at A, the signal from M must arrive at A with power at least higher than r_A. Given the distance between M and A, the transmit power of M must be at least,

$$P'_M \;>\; P_D \cdot \frac{G_D}{G_M} \cdot \left(\frac{d_{MA}}{d_{DA}}\right)^2, \tag{4.4}$$

where P_D is the transmit power of D and a LoS model is assumed for the channels between M and A to minimize P_M (least power requirement for the adversary). Similarly, to inject m_M at H, the transmit power of M must be at least,

$$P''_M \;>\; P_D \cdot \frac{G_D}{G_M} \cdot \left(\frac{d_{MH}}{d_{DH}}\right)^2. \tag{4.5}$$

From (4.4) and (4.5), to inject m_M simultaneously at A and H, the transmit power of M must be at least,

$$P_M \;>\; \max(P'_M, P''_M). \tag{4.6}$$

Let M perform an overshadowing attack during sweep $\mathbf{s}$ by transmitting at power P_M. The peak RSS ratio between the received signal at H and A is,

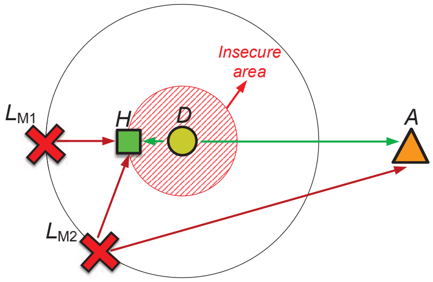


Figure 4.4: The adversary placed at L_{M1} (co-linear with H-to-A line) is the optimal position is outside the insecure area with fixed M-to-D distance for maximizing the RSS peak ratio.

$$\begin{aligned}
\max_{\mathbf{s}_M}(s_M(j)) &= \max_{\mathbf{r}_H}\left(\frac{r_H(j)}{r_A}\right) \\
&= \frac{G_H}{G_A}\max_{\mathbf{d}_{MH}}\left(\frac{d_{MA}}{d_{MH}(j)}\right)^2 \\
&= \frac{G_H}{G_A}\left(\frac{d_{MA}}{\min\limits_{\mathbf{d}_{MH}}(d_{MH}(j))}\right)^2 , \qquad (4.7)
\end{aligned}$$

where $\mathbf{s}_M$ is the sweep $\mathbf{s}$ affected by M's injection.

We investigate the optimal position of M that maximizes $\max_{\mathbf{s}_M}(s_M(j))$. From (4.7), $\max_{\mathbf{s}_M}(s_M(j))$ is maximized when d_{MA} is maximum while d_{MD} is minimum. Let us fix the distance d_{MD} to the smallest distance that M can maintain from D without being visually detected by the user. The position of M that maximizes $\max_{\mathbf{s}_M}(s_M(j))$ is achieved when M, H, D, and A are all co-linear in this particular order. This reflected in position L_{M1} (denoted by L_M^* from now) of Figure 4.4. At L_M^* the distance to A is maximized for a fixed distance to D, thus maximizing the achievable RSS ratio of M. The security analysis from here on assumes that M is at L_M^*. Fixing the position of M at L_M^*, we investigate the RSS ratio achieved at H under different helper motions. From Figure 4.6,

$$d_{MA} = d_{MD} + d_{DA},\ d_{MH} = (d_{MD} + d_{DH}\cos\theta)\sec\theta',$$

where θ corresponds to the angle between the D-to-H and D-to-A lines and θ' corresponds to the angle between the M-to-H and M-to-A lines. In addition, M achieves the maximum RSS ratio when H is closest to M. This corresponds to $\theta = 180°,\ \theta' = 0°$. In this case, (4.7) can be rewritten as,

$$\begin{aligned}\max_{\mathbf{s}_M}(s_M(j)) &= \frac{G_H}{G_A}\max_{\mathbf{d}_{DH}}\left(\frac{d_{MD} + d_{DA}}{(d_{MD} + d_{DH}(j)\cos\theta)\sec\theta'}\right)^2 \\ &= \frac{G_H}{G_A}\left(\frac{d_{MD} + d_{DA}}{d_{MD} - \max\limits_{\mathbf{d}_{DH}}(d_{DH}(j))}\right)^2 \\ &= \frac{G_H}{G_A}\left(\frac{d_{MD} + d_{DA}}{d_{MD} - d_{DH}^{\max}}\right)^2, \qquad (4.8)\end{aligned}$$

where $\max_{\mathbf{d}_{DH}}(d_{DH}(j)) = d_{DH}^{\max}$ which is the maximum distance between D and H during sweep $\mathbf{s}_M$. Using the value of τ_{peak} from (4.3) and (4.8), we evaluate the D-to-M distance outside which Test 1 detects a Type 1 adversary.

Proposition 3. *Test 1 detects any Type 1 adversary which is at distance d_{MD} from D, satisfying $(d_{MD}+d_{DA})^{\alpha_A}/(d_{MD}-d_{DH}^{\max})^2 < (d_{DA}/d_{DH}^{\min})^2$. Parameter α_A is the attenuation factor of the M-to-A channel.*

Proof. We first consider a single sweep of H over D. The threshold for detecting an invalid transmission using Test 1 is given by (**??**). This minimizes the τ_{peak} that needs to be met by the adversary to pass Test 1. Let the adversary be located at distance d_{MH} from H and d_{MA} from A. The RSS ratio achieved by a Type 1 adversary transmitting with power P_M is given by (4.1). As H is swept over D, the RSS ratios computed by A when M is active form a set $\mathbf{s}_M = \{s_M(1), s_M(2), \ldots, s_M(n)\}$, where $s_M(i)$ corresponds to the i^{th} RSS ratio sample collected by A while H moves over D. To detect a Type 1 adversary, it must follow that the maximum RSS ratio

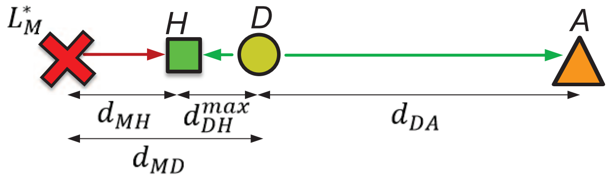


Figure 4.5: The optimal position L_M^* for defeating Test 1 when the distance d_{MD} is fixed, lies co-linearly with D and A.

obtained during the motion of H does not exceed τ_{peak}.

$$\begin{aligned}
\max_{\mathbf{s}_M}(s_M(i)) &< \tau_{peak} \\
\max_{d_{MA},\mathbf{d}_{MH}} \frac{G_H}{G_A}\cdot\frac{(d_{MA})^{\alpha_A}}{(d_{MH})^2} &< \tau_{peak} && \text{(4.9a)} \\
\frac{G_H}{G_A}\cdot\frac{(d_{MD}+d_{DA})^{\alpha_A}}{\max_{\mathbf{d}_{DH}}(d_{MD}-d_{DH}(j))^2} &< \tau_{peak} && \text{(4.9b)} \\
\frac{G_H}{G_A}\cdot\frac{(d_{MD}+d_{DA})^{\alpha_A}}{(d_{MD}-d_{DH}^{\max})^2} &< \frac{G_H}{G_A}\cdot\frac{(d_{DA})^2}{(d_{DH}^{\min})^2} && \text{(4.9c)} \\
\frac{(d_{MD}+d_{DA})^{\alpha_A}}{(d_{MD}-d_{DH}^{\max})^2} &< \frac{(d_{DA})^2}{(d_{DH}^{\min})^2}. && \text{(4.9d)}
\end{aligned}$$

In (4.9a), we replaced the expression of $s_M(i)$ from (4.7). In (4.9b), we considered the distances of M to H and A that maximize the RSS ratio. We further fixed the distance between M and D and considered all possible locations of M relative to H and A that maximize the RSS ratio achieved by M. This occurs when M, D, H, and A to be co-linear as shown in Figure 4.5. In (4.9c), we further minimized the denominator by considering the location of H closest to M.

The sweeping motion of H over D is repeated multiple times. The maximum RSS ratio must exceed τ_{peak} for every motion of H. Given that the sweeps are of approximately equal length, the same inequality as in (4.9d) must be satisfied for all sweeps. This concludes the proof.

□

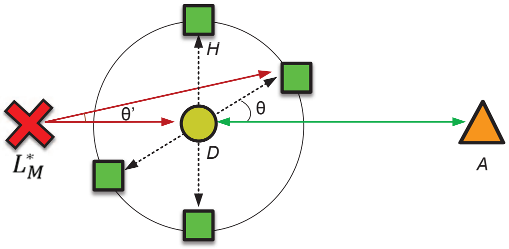


Figure 4.6: Various motions for the helper.

When $\alpha_A = 2$, there is only one positive solution for d_{MD} condition stated in Proposition 1. The adversary has to get closer to H than D is to defeat Test 1. When $\alpha_A > 2$ there are two solutions to the condition of Proposition 1. For instance, let us consider $\alpha_A = 4$. By solving for d_{MD}, we find that a Type 1 adversary is successful in two regions

$$d_{MD} < {}^{d_{DA} - 2d_{DA}d_{DH}^{\min} + \sqrt{d_{DA}(d_{DA} - 4d_{DA}d_{DH}^{\min} - 4d_{DH}^{\min}d_{DH}^{\max})}}\big/_{2d_{DH}^{\min}},$$

$$d_{MD} > {}^{2d_{DA}d_{DH}^{\min} - d_{DA} + \sqrt{d_{DA}(d_{DA} - 4d_{DA}d_{DH}^{\min} - 4d_{DH}^{\min}d_{DH}^{\max})}}\big/_{2d_{DH}^{\min}}.$$

The first inequality is similar to the case of $\alpha_A = 2$. The adversary has to get close enough to H and D to defeat Test 1. The second inequality however, reveals the interesting case where if M moves far away from H and A, he will eventually achieve an RSS ratio higher than τ_{peak}. This is intuitive because the signal attenuates faster to A than to H due to the higher path loss exponent for the M-to-A channel. However, moving away from the legitimate devices poses a high power requirement for succeeding in the overshadowing attack. According to (4.6), the power required for a successful overshadowing attack grows as a function of $(d_{MA})^{\alpha_A}$. For example, for $d_{DA} = 8$m, $d_{DH}^{\min} = 4$cm, $d_{DH}^{\max} = 8$cm, $P_D = 1$mW and $G_D = G_M = 1$, the D-to-M distances $d_{MD} < 4.1$cm, or $d_{MD} > 174$m for $\alpha_A = 3$, and $d_{MD} < 4.12$cm, or $d_{MD} > 230$m for $\alpha_A = 4$. For the longer d_{MD} solutions, the transmission power to successfully launch overshadowing attacks is $P_M = 18.9$KW for $\alpha_A = 3$ and $P_M = 33$KW for $\alpha_A = 4$, which are prohibitive.

Type 2 adversary: A Type 2 adversary can independently control the received power at H and A using directional antennas. To defeat Test 1, an adversary has to achieve $\max_{\mathbf{s}_M}(s_M(j)) \geq \tau_{peak}$ when

$$\max_{\mathbf{s}_M}(s_M(j)) = \max_{\mathbf{r}_H}\left(\frac{r_H(j)}{r_A}\right) \tag{4.10a}$$

$$= \frac{P_H G_H G_{MH}}{P_A G_A G_{MA}}\left(\frac{d_{MA}}{d_{MD} - d_{DH}^{\max}}\right)^2, \tag{4.10b}$$

where P_X is the power transmitted from M to X, G_{MX} is the directional antenna gain of M transmitting to X. Without loss of generality in (4.10b), we assume LoS channels from M to any other device (our goal is to show some scenario for which a Type 2 adversary defeats Test 2). From (4.10b), an adversary achieves $\max_{\mathbf{s}_M}(s_M(j)) \geq \tau_{peak}$ when,

$$\frac{P_H}{P_A} \geq \tau_{peak}\frac{G_A G_{MA}}{G_H G_{MH}} \cdot \left(\frac{d_{MD} - d_{DH}^{\max}}{d_{MA}}\right)^2. \tag{4.11}$$

The condition of (4.11), dictates the adversary's strategy for defeating Test 1. By controlling the powers of directional transmissions P_H and P_A, he can achieve an RSS ratio that exceeds τ_{peak}. One trivial strategy is to choose a low P_A such that (4.11) is satisfied. However note that P_A must be sufficiently large to carry out an overshadowing attack at A, as given by (4.5). To detect a Type 2 adversary, we introduce Test 2 that checks the dynamic RSS ratio range.

4.4.2 Test 2: RSS Ratio Dynamic Range

In the second test, the hub computes the dynamic range of the RSS ratio for each sweep $\mathbf{s}_i$ as:

$$\Delta_i = \frac{\max\limits_{\mathbf{s}_i} s(i,j)}{\min\limits_{\mathbf{s}_i} s(i,j)}.$$

The device passes Test 2 if $\Delta_i \geq \tau_{range}$, for every $\mathbf{s}_i$. This test exploits the higher roll-off rate of the signal power at short distances relative to longer ones. An adversary transmitting a few meters away from H invokes a smaller dynamic range than that of D. Formally, Test 2 has following steps:

1. **Dynamic range computation:** For a sweep set $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_\ell$, A computes Δ_i as,
$$\Delta_i = \frac{\max\limits_{\mathbf{s}_i} s(i,j)}{\min\limits_{\mathbf{s}_i} s(i,j)}, \ \forall\ i = 1, 2, \ldots, \ell.$$
2. **Dynamic range test:** If $\Delta_i \geq \tau_{range}$, $\forall\ i = 1, 2, \ldots, \ell$, then D passes Test 2.

Determining τ_{range}: Similar to Test 1 and without loss of generality, we focus on a single sweep $\mathbf{s}$. The RSS ratio range measured when a legitimate device D transmits is:

$$\begin{aligned}\Delta &= \frac{\max\limits_{\mathbf{s}} s(j)}{\min\limits_{\mathbf{s}} s(j)} \\ &= \frac{\max\limits_{\mathbf{r}_H} \left({}^{r_H(j)}\!/_{r_A}\right)}{\min\limits_{\mathbf{r}_H} \left({}^{r_H(j)}\!/_{r_A}\right)} = \frac{\max\limits_{\mathbf{r}_H} r_H(j)}{\min\limits_{\mathbf{r}_H} r_H(j)} \\ &= \left(\frac{\max\limits_{\mathbf{d}_{DH}}(d_{DH}(j))}{\min\limits_{\mathbf{d}_{DH}}(d_{DH}(j))}\right)^2 = \left(\frac{d_{DH}^{\max}}{d_{DH}^{\min}}\right)^2, \end{aligned} \tag{4.12}$$

where $d_{DH}^{\max}$ and $d_{DH}^{\min}$ are the maximum and minimum distances between D and H during sweep $\mathbf{s}$. In (4.12), the channel from D-to-A was assumed to be constant during the sweep $\mathbf{s}$ (and hence r_A at the nominator and the denominator is cancelled) and a LoS channel was assumed for the D-to-H channel ($\alpha_H = 2$) due to the proximity between D and H (within a few cm). The value of τ_{range} is selected to be equal to Δ, given conservative estimates for the user's range of motion during sweeps.

$$\tau_{range} \leq \Delta = {}^{d^{\max}_{DH}}\!/\!_{d^{\min}_{DH}}{}^{2}. \tag{4.13}$$

Note that considering a pathloss exponent equal to $\alpha_H = 2$ for the D-to-H channel yields the most conservative value for τ_{range}. A legitimate device with $\alpha_H > 2$, will pass Test 2 when τ_{range} is set according to $\alpha_H = 2$, as shown in (4.13). We evaluate the real-world threshold values for various parameters and user motions in Section 4.5.2. We use the value of τ_{range} in (4.13) to evaluate the capability of Type 2 and Type 3 adversaries in defeating Test 2.

Security Analysis:

Type 2 adversary: A Type 2 adversary can control the received powers at H and A independently via directional transmissions. To defeat Test 2, the adversary has to achieve $\Delta_M \geq \tau_{range}$. The RSS ratio dynamic range depends on the motion of H given that A is static:

$$\begin{aligned}
\Delta_M &= \frac{\max\limits_{\mathbf{s}_M} s_M(j)}{\min\limits_{\mathbf{s}_M} s_M(j)} \\
&= \frac{\max\limits_{\mathbf{r}_H} \left({}^{r_H(j)}\!/\!_{r_A}\right)}{\min\limits_{\mathbf{r}_H} \left({}^{r_H(j)}\!/\!_{r_A}\right)} && \text{(4.14a)} \\
&= \left(\frac{\max\limits_{\mathbf{d}_{DH}}((d_{MD} + d_{DH}(j)\cos\theta)\sec\theta')}{\min\limits_{\mathbf{d}_{DH}}((d_{MD} + d_{DH}(j)\cos\theta)\sec\theta')} \right)^{\alpha_H} && \text{(4.14b)} \\
&= \left(\frac{d_{MD} + d^{\max}_{DH}}{d_{MD} - d^{\max}_{DH}} \right)^{\alpha_H}, && \text{(4.14c)}
\end{aligned}$$

where α_H is the attenuation factor of the M-to-H channel which is constant during the sweep. In (4.14a), the largest and smallest RSS ratio samples over all sweep samples were considered. In (4.14b), the min and max relative distance between M and H was considered in the general channel model with a pathloss exponent α_H, and in (4.14c), the maximum Δ_M was derived assumed an optimal orientation for M where $\theta = \theta' = 0°$ (farthest from M) for the numerator and $\theta = 180°,\ \theta' = 0°$ (closest to M) for the denominator.

Although the attenuation factor α_H could vary in different settings, a directional attack targeting both H and A should have LoS to H and A, so that the adversary can aim at the two devices. In the next proposition, we compute the D-to-M distance for a Type 2 adversary when ($\alpha_H = 2$).

Proposition 4. *A Type 2 adversary is detected by Test 2 when* $d_{MD} > {}^{d_{DH}^{\max}(d_{DH}^{\max}+d_{DH}^{\min})}\!/\!{}_{d_{DH}^{\max}-d_{DH}^{\min}}$.

Proof. Without loss of generality due to the similar nature of every sweep, consider a single sweep $\mathbf{s}$ of H over D. The true RSS ratio range (Δ) for a legitimate device D is given by (4.12). The value of τ_{range} is selected to be equal to Δ, given conservative estimates for the user's range of motion during sweeps.

The RSS ratio range (Δ_M) for a Type 2 adversary that is active during a sweep $\mathbf{s}_M$ is given by (4.14c).

To pass Test 2,

$$\Delta_M < \tau_{range} \tag{4.15a}$$

$$\left(\frac{d_{MD} + d_{DH}^{\max}}{d_{MD} - d_{DH}^{\max}}\right)^2 < \left(\frac{d_{DH}^{\max}}{d_{DH}^{\min}}\right)^2 \tag{4.15b}$$

$$\left(\frac{d_{MD} + d_{DH}^{\max}}{d_{MD} - d_{DH}^{\max}}\right) < \left(\frac{d_{DH}^{\max}}{d_{DH}^{\min}}\right) \tag{4.15c}$$

$$(d_{MD} + d_{DH}^{\max})\, d_{DH}^{\min} < (d_{MD} - d_{DH}^{\max})\, d_{DH}^{\max} \tag{4.15d}$$

$$d_{MD} d_{DH}^{\min} + d_{DH}^{\max} d_{DH}^{\min} < d_{MD} d_{DH}^{\max} - (d_{DH}^{\max})^2 \tag{4.15e}$$

$$d_{MD} d_{DH}^{\max} - d_{MD} d_{DH}^{\min} > d_{DH}^{\max}(d_{DH}^{\max} + d_{DH}^{\min}) \tag{4.15f}$$

$$d_{MD}(d_{DH}^{\max} - d_{DH}^{\min}) > d_{DH}^{\max}(d_{DH}^{\max} + d_{DH}^{\min}) \tag{4.15g}$$

$$d_{MD} > \frac{d_{DH}^{\max}(d_{DH}^{\max} + d_{DH}^{\min})}{d_{DH}^{\max} - d_{DH}^{\min}} \tag{4.15h}$$

In (4.15c), it is assumed that $d_{MD} > d_{DH}^{\max}$. The inequality in (4.15h) yields the distance of M from D after which a Type 2 adversary is detectable by Test 2. Note that for nominal user motions it holds that $d_{DH}^{\max} >> d_{DH}^{\min}$) in which case $d_{MD} > d_{DH}^{\max}$. That is the adversary, becomes detectable if it is at a distance longer than the boundary of H's motion. This concludes the proof. □

For $\alpha_H > 2$ the D-to-M distance decreases. For example, if $\alpha_H = 4$, then

$$d_{MD} \leq (d_{DH}^{\max})^2 - {}^{(d_{DH}^{\min} d_{DH}^{\max} - 2(d_{DH}^{\min})^2 \sqrt{(d_{DH}^{\max}/d_{DH}^{\min})^3})}\big/{}_{(d_{DH}^{\min} + d_{DH}^{\max})}$$

or approximately $d_{MD} < (d_{DH}^{\max})^2$. However, under such a multipath environment, it is difficult to direct the power of a directional transmission to a single target. Some power is inevitably received by A thus decreasing the RSS ratio dynamic range. Moreover, a higher pathloss exponent increases by several orders of magnitude the power necessary to launch a successful overshadowing attack. Even if $d_{MD} < (d_{DH}^{\max})^2$, the D-to-M distance remains relatively large given the short distance between H and D (20 cm in our experiments).

Type 3 adversary: A Type 3 adversary can apply fine-grained power control during a sweep. To defeat Test 2, the adversary has to achieve $\Delta_M \geq \tau_{range}$. To do so, the adversary can manipulate the $\max_{\mathbf{r}_H} \left({}^{r_H(j)}/_{r_A}\right)$ and $\min_{\mathbf{r}_H} \left({}^{r_H(j)}/_{r_A}\right)$ within a sweep $\mathbf{s}_M$ by regulating the power received by A and H, respectively. The maximum RSS ratio in $\mathbf{s}_M$ is given by

$$\begin{aligned} \max_{\mathbf{s}_M} s_M(j) &= \max_{\mathbf{r}_H} \frac{r_H(j)}{r_A} \\ &= \frac{P_H G_H G_{MH}}{P_A G_A G_{MA}} \left(\frac{d_{MA}}{\min\limits_{\mathbf{d}_{DH}} (d_{MD} + d_{DH}(j) \cos\theta) \sec\theta'} \right)^2 && (4.16a) \\ &= \frac{P_H G_H G_{MH}}{P_H G_A G_{MA}} \left(\frac{d_{MA}}{d_{MD} - d_{DH}^{\max}} \right)^2, && (4.16b) \end{aligned}$$

where P_H, P_A are the transmission powers from M to H and from M to A respectively and G_{MH}, G_{MA} are the directional antenna gains from M to H and M to A, respectively. In (4.16a), we have used a LoS channel to maximize the RSS ratio assuming a fixed channel to A. In (4.16b), we further set the orientation of M to $\theta = 180°$, $\theta' = 0°$ to minimize the denominator. Using similar arguments, the minimum RSS ratio is achieved when:

$$\begin{aligned} \min_{\mathbf{s}_M} s_M(j) &= \min_{\mathbf{r}_H} \frac{r_H(j)}{r_A} \\ &= \frac{P'_H G_H G_{MH}}{P'_A G_A G_{MA}} \left(\frac{d_{MA}}{\max\limits_{\mathbf{d}_{DH}} (d_{MD} + d_{DH}(j) \cos\theta) \sec\theta'} \right)^2 && (4.17a) \\ &= \frac{P'_H G_H G_{MH}}{P'_A G_A G_{MA}} \left(\frac{d_{MA}}{d_{MD} + d_{DH}^{\max}} \right)^2, && (4.17b) \end{aligned}$$

where P'_H, P'_A are the transmission powers of M to H and A respectively, which can differ from the powers used when the max RSS ratio is achieved. In (4.17a), we have used $\alpha = 2$ for the respective channels from M to H and M to A without loss of generality, as we only need to demonstrate that under certain conditions, a Type 3 adversary can defeat Test 2. We further set the orientation of M to $\theta = \theta' = 0°$ to maximize the denominator. From (4.16b) and (4.17b), the RSS ratio range for sweep $\mathbf{s}_M$ is given by:

$$\Delta_M = \frac{P_H P'_A}{P'_H P_A} \left(\frac{d_{MD} + d_{DH}^{\max}}{d_{MD} - d_{DH}^{\max}} \right)^2 \approx \frac{P_H P'_A}{P'_H P_A}, \tag{4.18}$$

where for sufficiently large d_{MD} we have approximated $d_{MD} - d_{DH}^{\max} \approx d_{MD} + d_{DH}^{\max} \approx d_{MD}$.

From (4.13) and (4.18) we derive the condition under which the adversary defeats Test 2 from any location as

$$\frac{P_H P'_A}{P'_H P_A} \geq \tau_{range}. \tag{4.19}$$

The condition in (4.19), dictates the adversary's strategy for defeating Test 2. By controlling the ratios $^{P_H}/_{P'_H}$ and $^{P'_A}/_{P_A}$, he can achieve a desirable dynamic range that exceeds τ_{range}. The latter is defined by the distance ratio $\left(^{d_{DH}^{\max}}/_{d_{DH}^{\min}}\right)^2$. For an effective attack, it is expected that $P_H > P'_H$ whereas $P'_A > P_A$ such that the product of the ratios becomes large. One may trivially assume that choosing very low values for P'_H and P_A is sufficient to exceed τ_{range}. However, the powers selected by M for each directional transmission are lower-bounded by the minimum power required to carry out overshadowing attack at A and H, as dictated by (4.5) and (4.4), respectively. To detect a Type 3 adversary, we introduce Test 3 that checks the time period of every sweep.

4.4.3 Test 3: Sweep Period

In the third test, the hub measures the period $T(i)$ of each sweep $\mathbf{s}_i$ to verify the sweep consistency. The main idea here is that the user takes approximately the same time to complete a sweep. We fist define the sweep period $T(i)$.

Definition 2. *Sweep period $T(i)$:* The sweep period $T(i)$ of sweep $\mathbf{s}_i$ is defined as the time difference between the occurring times of the first and last sweep sample.

Sweep Consistency is verified by checking if the ratio ${}^{T(i)}\!/_{T(j)} \leq \tau_{period}$, $\forall\, i, j; i \neq j$ where the longer period is always placed at the nominator. Formally, Test 3 has following steps:

1. **Sweep period computation:** For a sweep set $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_\ell$, the sweep period $T(i)$ corresponding to the sweep $\mathbf{s}_i$ is computed according to Definition 2.
2. **Sweep period test:** If

$$ {}^{T(i)}\!/_{T(j)} \leq \tau_{period} \;\forall\; i, j;\;\; i \neq j $$

then D passes Test 3.

Figure 4.7(a) shows the helper's locations where RSS ratio peaks and valleys are observed during a sweep $\mathbf{s}_i$ for two motions, when D is transmitting. It is expected that these periods would be fairly consistent given that H passes over D with every motion and the speed of motion is relatively constant. On the other hand, the sweep periods when a transmission originates from a remote location do not present the same consistency. For a subset of helper motions, the sweep period takes twice as long because the helper does not pass over the remote device. This is demonstrated in Figure 4.7(b) where a Type 3 adversary is performing an overshadowing attack at H from a remote location L_M^*. We have divided the area where the helper moves into two areas X and Y. These areas are defined by the intersection of two circles. The circle C_1 is centered at D and has a radius $d_{DH}^{\max}$. The circle C_2 is centered at L_M^* and has a radius d_{MD}. Assuming a straight line movement, when the helper's motion ends in the boundaries of Y (e.g., horizontal motion), the distance between two helper locations where two consecutive peaks occur is two times the disk diameter (when the helper reaches the disk boundary closest to L_M^*). For a motion that ends in the boundaries of X (e.g., vertical motion), two consecutive peaks occur after a

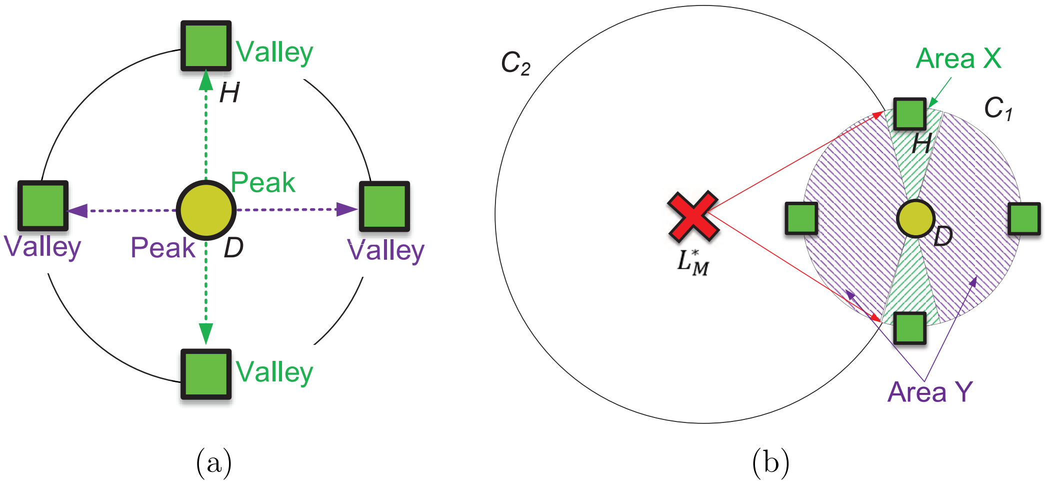


Figure 4.7: (a) Peaks and valleys of Γ for various movement of H when D is transmitting, and (b) a sweep in area Y takes at least twice as much as a sweep in area X when M is transmitting from a remote location.

distance of at most one diameter. The third test exploits this irregularity in the sweep periods to detect a remote attack.

Determining τ_{period}: For a legitimate device D, the time to complete sweep $\mathbf{s}_i$ is given by,

$$T(i) = \frac{2d_{DH}^{\max}(i)}{v}, \tag{4.20}$$

where v is the average sweep speed and $d_{DH}^{\max}(i)$ is the maximum distance between D and H in the i^{th} sweep. The ratio of the sweep periods ($\mathbf{s}_i$ and $\mathbf{s}_j$) when D is transmitting is given by,

$$\frac{T(i)}{T(j)} = \left(\frac{2d_{DH}^{\max}(i)}{v}\right)\left(\frac{v}{2d_{DH}^{\max}(j)}\right) = \frac{d_{DH}^{\max}(i)}{d_{DH}^{\max}(j)}, \tag{4.21}$$

where the average sweep speed is assumed to be relatively the same between sweeps. To pass the Test 3, we can select

$$\tau_{period} \leq {}^{T(i)}/_{T(j)} = {}^{d_{DH}^{\max}(i)}/_{d_{DH}^{\max}(j)}. \tag{4.22}$$

The threshold depends on the helper motion. The user is given instructions to perform consistent sweeps in range and speed so that ${}^{T(i)}/_{T(j)} \approx 1$. However, to allow for a margin of error in the user's motions the threshold τ_{period} can be set to a value between 1 and 2. In Section 4.5.3, we experimentally show that a selection of $\tau_{period} = 1.4$ provides a sufficient error margin for motion variation.

Security Analysis: We now analyze the ability of a Type 3 adversary in defeating Test 3. Note that a Type 3 adversary incorporates the Type 1 and Type 2 capabilities and therefore a successful test will defend against all three adversaries. To defeat Test 2, a Type 3 adversary applies power control to achieve the desired RSS ratio dynamic range τ_{range}. This is achieved by injecting a maximum power P_H when H is the closest to M and a minimum power P'_H when H is farthest from M thus maximizing the range achieved measured by H. The sweep period recorded by H when M is active depends on the trajectory of H relative to M's location L^*_M. This period is defined as the time between two successive RSS ratio peaks, which are achieved when the M-to-H distance $d_{MH}(i)$ becomes minimum. Analyzing the geometry of Figure 4.7(b), minimum of $d_{MH}(i)$ is achieved either on the perimeter of area Y (when H's motion terminates in Y) or inside the area of Y closest to L^*_M. In the first case, the sweep period is the time required to traverse a distance equal to twice the range of H's motion, whereas in the latter case the sweep period is the time required to traverse a distance at most one time H's range of motion. Using the threshold τ_{period} from (4.22) and the adversary's sweep period, the success of Test 3 is expressed in the following proposition.

Proposition 5. *A Type 3 adversary is always detected by Test 3 if (a) the user performs at least two sweeps* $\mathbf{s}_1$ *and* $\mathbf{s}_2$*, (b) sweep* $\mathbf{s}_1$ *starts and ends in area* X*, whereas sweep* $\mathbf{s}_2$ *starts and ends in area* Y*, and (c)* $d_{MD} > \min_i(d^{\max}_{DH}(i)) \ \forall\ i = 1, \ldots, \ell$*.*

Proof. To defeat Test 2, a Type 3 adversary applies power control to achieve the desired RSS ratio dynamic range τ_{range}. This is achieved by injecting a maximum power P_H when H is the closest to M and a minimum power P'_H when H is farthest

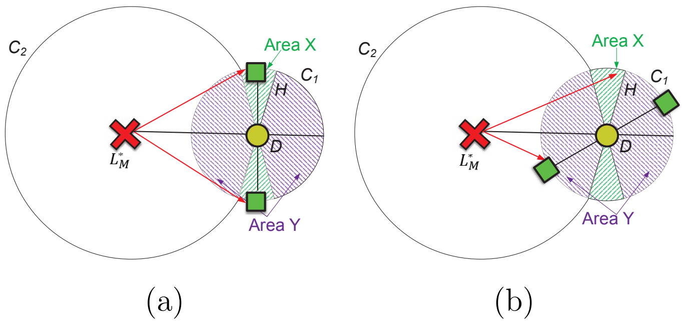


Figure 4.8: (a) Sweep with H starting and ending in area X, (b) Sweep with H starting and ending in area Y.

from M thus maximizing the range achieved measured by H. However, we show that this approach leads to a violation of Test 3.

Consider two sweeps $\mathbf{s}_1$ and $\mathbf{s}_2$ performed by the user. The ratio of the sweep periods is given by

$$\frac{T(i)}{T(j)} = \frac{d_{DH}^{\max}(i)}{d_{DH}^{\max}(j)}, \tag{4.23}$$

where we have assumed a constant average speed for both $\mathbf{s}_1$ and $\mathbf{s}_2$. Let the threshold for passing Test 3 be set to $\tau_{period} = {d_{DH}^{\max}(i)}/{d_{DH}^{\max}(j)}$. Under equal sweep lengths, this ratio is equal to one[1]. Let the area where H moves around D be divided into two sub-areas X and Y, as shown Figure 4.8. The sub-areas are defined by the intersection of two circles. Sub-area Y consists of the sector S_1 formed by the intersection between C_1 and C_2 and the sector of C_1 that is symmetric to S_1 over the y-axis. Sub-area X is the complement of sub-area Y within C_1.

Let the adversary perform its power control attack during sweeps $\mathbf{s}_1$ and $\mathbf{s}_2$. Let also $\mathbf{s}_1 \in X$ and $\mathbf{s}_2 \in Y$. From the sub-area geometry, it follows that any sweep (sweeps are assumed to form a straight line) that originates in X ends in X and any sweep that originates in Y ends in Y. Moreover, for any point $R \in X$, it follows that $d_{MR} \geq d_{MD}$. Therefore, when a sweep is performed in X, the D is the closest point to L_M^* (also the point where M transmits with P_H.) In this case, the sweep period, i.e., time between two successive peaks, equals the time until two successive

[1] In reality, the sweep lengths are unequal but approximately the same. Based on the experiments presented in the evaluation section, we have found the $\tau_{period} = 1.4$. That is, the sweep period can vary as much as 40%.

visits of H over D. Equivalently, this is equal to the time required to travel the diameter of circle C_1, denoted by $d_{DH}^{\max}$.

On the other hand. for any sweep in Y, it is straightforward to show from the sub-area geometry that the minimum separation between M and H is achieved when H is the farthest from D (i.e., at a point in C_1). In this case, the sweep period, equals the time until two successive visits of H at maximum separation from D. Equivalently, this is equal to the time required to travel *two times* the diameter of circle C_1, denoted by $2d_{DH}^{\max}$. Assuming a constant average speed of motion for H, the ratio ${}^{T(i)}/{}_{T(j)} = 2 > \tau_{period}$. Therefore, a violation of Test 3 will be recorded. If the two sweeps $\mathbf{s}_1$ and $\mathbf{s}_2$ belong to the same sub-area, the same sweep period will be recorded for both of them and a Type 3 adversary will pass Test 3.

Note that a Type 3 adversary incorporates the Type 1 and Type 2 capabilities and therefore a successful test will defend against all three adversaries.

This concludes the proof. □

In the proposition, we perform the evaluation for D-to-M distance as assumed in the other tests. For the secure region, the sweep period achieved by the adversary is ${}^{T_M(i)}/{}_{T_M(j)} \geq 2$. However, for outside the secure region $d_{DH}^{\max} \leq d_{MD} < 0$, the sweep period ratio is $2 < {}^{T_M(i)}/{}_{T_M(j)} < 1$, which can give ambiguous detection as the sweep period of the device varies between 1.4 and 1.

Test 3 relies on the user to perform specific motions, which may not always be performed. To disassociate the security of the RSS authenticator from the sweep motion orientation, we introduce Test 4.

4.4.4 Test 4: RSS Ratio and Motion Correlation

In the fourth test, the hub correlates the helper motion with the RSS ratio fluctuation. This test requires acceleration data from H to identify the beginning and end of a sweep, independent of Γ. During the sweeping motion over the legitimate device D, the helper changes direction at its maximum separation from D. This change in direction causes a peak in the acceleration of H and a valley in Γ. For this test, we define the minimum RSS ratio instances as:

Definition 3. *Minimum RSS ratio instances* $\mathbf{t}_{RSS}$*:* Let $\mathcal{F}$ be the fitted smooth curve on the RSS ratio sample set Γ. The set of minimum RSS ratio instances $\mathbf{t}_{RSS} = \{t_{RSS}(1), t_{RSS}(2), \ldots, t_{RSS}(\ell)\}$ is a set of times corresponding to the local RSS ratio minima in $\mathcal{F}$.

In addition, we define the motion change instances as:

Definition 4. *Motion change instances* $\mathbf{t}_{acc}$*:* Let $\mathbf{a}$ be the set of acceleration values of H during the sweeping motion ordered according to time. Let $\mathcal{F}_{\mathbf{a}}$ be the fitted smooth curve on $\mathbf{a}$. The motion change instances $\mathbf{t}_{acc} = \{t_{acc}(1), t_{acc}(2), \ldots, t_{acc}(\ell)\}$ is the set of times corresponding to the local minima in $\mathcal{F}_{\mathbf{a}}$.

A device passes Test 4 if the root mean square error (RMSE) between $\mathbf{t}_{RSS}$ and $\mathbf{t}_{acc}$ is below a threshold τ_{corr}. This test particularly targets a Type 3 adversary who may defeat Tests 1 and 2 via fine-grained power control or if Test 3 does not include the necessary helper motions that yield different motion periods. If the adversary cannot synchronize the power fluctuation with the helper motion, which is difficult to achieve in real time, the fourth test is violated. Formally, Test 4 has following steps:

1. **Acceleration data transmission:** The helper sends the minimum RSS ratio and motion change instances $\mathbf{t}_{RSS}$, $\mathbf{t}_{acc}$ to the hub, using AE($\cdot$) (authenticated encryption),

2. **RMSE calculation** The hub A computes the root mean square error (RMSE) between $\mathbf{t}_{RSS}$ and $\mathbf{t}_{acc}$ as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{\ell}(t_{RSS}(i) - t_{acc}(i))^2}{\ell}}$$

3. **RSS ratio–motion correlation test:** If $RMSE \leq \tau_{corr}$, D passes Test 4.

Determining τ_{corr}: Because the helper has a LoS channel to D and the distance between D and A is fixed, the RSS ratio is proportional to the distance between H and D. Therefore, the minimum RSS ratio is achieved at the largest separation between H and D, which is also the point of maximum acceleration as the legitimate device is changing direction. However, variation in RSS and the perturbations introduced by the user motion can lead to a time misalignment between $\mathbf{t}_{acc}$ and $\mathbf{t}_{RSS}$. Let the mean time misalignment between any two samples $t_{acc}(i)$ and $t_{RSS}(i)$ be bounded by $|t_{acc}(i) - t_{RSS}(i)| \leq \epsilon$. The RMSE can then be bounded to

$$RMSE = \sqrt{\frac{\sum_{i=1}^{\ell}(t_{RSS}(i) - t_{acc}(i))^2}{\ell}} \leq \sqrt{\frac{\ell\epsilon^2}{\ell}} = \epsilon. \tag{4.24}$$

We can set τ_{corr} for passing Test 4 to a value slightly larger than ϵ. We have experimentally evaluated the mean time misalignment error ϵ in Section 4.6.

Security Analysis: To defeat Test 4, a Type 3 adversary has to achieve $RMSE \leq \tau_{corr}$, for all the sweeps. This can be done by applying power control and synchronizing the power variation with the motion of H in real time. That is, M must predict the acceleration peaks (at the edges of the user's motion) and force RSS valleys at those locations. One can consider that this condition can be satisfied without power control if the adversary selects his location such that the M-D line is perpendicular to the helper motion. However, the helper is moved over D in more than one orientations so there is no one location (other then D's location) that satisfies this criterion. Therefore, the adversary has to apply power control in real time to match the RSS valleys with the acceleration peaks.

Assuming that the helper motion cannot be directly observed and analyzed in real time (via a camera system), the adversary can attempt to synchronize the power control by guessing the average motion period T and the motion start time. Consider the series $\mathbf{t}_{acc}$ recorded by the helper as a time reference. Let the adversary vary the RSS power at H using a period $\mathcal{T}$. The error between any two samples $t_{acc}(i)$ and $t_{RSS}^{(M)}(i)$ be bounded by $|t_{acc}(i) - t_{RSS}^{(M)}(i)| \leq i\mathbf{\Delta} + \boldsymbol{\epsilon}_M$. Where $\mathbf{\Delta}$ is the random variable depicting error induced in estimating sweep period by the adversary and $\boldsymbol{\epsilon}_M$ is a random variable depicting the misalignment between the acceleration peaks and RSS valleys due to the unknown motion start time. Note that the value of $\boldsymbol{\Delta}$ is not affecting the RMSE for the legitimate device because the peaks and values are recorded at the edge of the motion, even if the motion period changes. For an adversary varying the RSS with a fixed period, on the other hand, the error caused by $\boldsymbol{\Delta}$ accumulates with the number of sweeps. Moreover, the misalignment error $\boldsymbol{\epsilon}_M$ is expected to be much larger than ϵ, because the start time of the user's motion is unknown. In the next proposition, we explore the number of minimum sweeps ℓ^* required to detect a Type 3 adversary using Test 4.

Proposition 6. *Test 4 detects a Type 3 adversary with probability no smaller than p_0, when the user performs at least*

$$\ell^* \geq \max\left[1, \left\lceil \frac{\sqrt{1 + {}^{48\epsilon^2}\!/\!{}_{\delta^2(1-p_0)^2}} - 3}{4} \right\rceil\right]$$

sweeps, the sweep period estimation error is uniformly distributed in $[-\delta, \delta]$, and the threshold for passing Test 4 is set to ϵ.

Proof. To pass Test 4, a Type 3 adversary must synchronize the RSS ratio minima measured by the helper with the acceleration maxima. Let $\mathbf{t}_{acc} = \{t_{acc}(1), t_{acc}(2), \ldots, t_{acc}(\ell)\}$ be the times where H records its maximum acceleration (at the ends of each sweep motion) and let $\mathbf{t}^M_{RSS} = \{t^M_{RSS}(1), t^M_{RSS}(2), \ldots, t^M_{RSS}(\ell)\}$ by the times where M induces the RSS ratio minima at H via directional transmissions and power control. The adversary M must select $\mathbf{t}^M_{RSS}$ such that it matches the periodicity of $\mathbf{t}_{acc}$. However, there are two sources of error that make this matching difficult. First, the period of the helper's motion is not fixed due to the variation induced by the user's hand motion. Second, the start time of the motion is not known unless it is directly observed with a high accuracy camera system. The latter is a very strong requirement that would reveal the presence of an adversary. We capture the two sources of error between $\mathbf{t}_{acc}$ and $\mathbf{t}_{RSS}^{M}$ in the following relationship:

$$|t_{acc}(i) - t_{RSS}^{(M)}(i)| = i\mathbf{\Delta} + \mathbf{E}_M$$

where $\mathbf{\Delta}$ is a random variable denoting the estimation error for the period of each sweep and $\mathbf{E}_M$ is a random variable denoting the misalignment between the acceleration peaks and RSS ratio valleys due to the unknown motion start time. Note that the error for the sweep period is cumulative with every sweep, whereas the start time error is only at the beginning of the motion. For Test 4, the RMSE achieved by the adversary becomes,

$$\begin{aligned} RMSE_M &= \sqrt{\frac{\sum_{i=1}^{\ell}(t_{acc}(i) - t_{RSS}^{(M)}(i))^2}{\ell}} \\ &= \sqrt{\frac{\sum_{i=1}^{\ell}(i\mathbf{\Delta} + \mathbf{E}_M)^2}{\ell}}, \end{aligned} \tag{4.25}$$

where ℓ is the number of sweeps. We now show that even if the the adversary knows the motion starting time ($\mathbf{E}_M = 0$), the error in the sweep period estimation will make him fail the test, given sufficient number of sweeps. For this worst case ($\mathbf{E}_M = 0$),

$$\begin{aligned} RMSE_M &= \sqrt{\frac{\sum_{i=1}^{\ell}(i\mathbf{\Delta})^2}{\ell}} \\ &= |\mathbf{\Delta}|\sqrt{\frac{\sum_{i=1}^{\ell} i^2}{\ell}} \\ &= |\mathbf{\Delta}|\sqrt{\frac{(\ell+1)(2\ell+1)}{6}}. \end{aligned} \tag{4.26}$$

Without loss of generality, let $\mathbf{\Delta}$ by uniformly distributed in $[-\delta, \delta]$. We analyze this case here because of the simple form of the distribution for $|\mathbf{\Delta}|$, but the latter is computable for any distribution. For a uniformly distributed $\mathbf{\Delta}$, the PDF of the $RMSE_M$ is uniformly distributed in $[0, \delta\sqrt{\frac{(\ell+1)(2\ell+1)}{6}}]$. This easily follows from eq. (4.26) and the fact the $|\boldsymbol{\Delta}|$ is uniformly distributed in $[0, \delta]$. Test 4 detects an adversary if $RMSE_M$ exceeds ϵ. Given that $RMSE_M$ is a random variable, we calculate the probability that it exceeds ϵ using the CDF.

$$\Pr[RMSE_M > \epsilon] = 1 - \frac{\epsilon}{\delta\sqrt{\frac{(\ell+1)(2\ell+1)}{6}}}. \tag{4.27}$$

We calculate the minimum number of sweeps ℓ^* required such that $RMSE_M$ exceeds ϵ with probability at least p_0.

$$\begin{aligned}
\Pr[RMSE_M > \epsilon] &\geq p_0, &\text{(4.28a)}\\
1 - \frac{\epsilon}{\delta\sqrt{\frac{(\ell+1)(2\ell+1)}{6}}} &\geq p_0, &\text{(4.28b)}\\
2\ell^2 + 3\ell + 1 &\geq \frac{6\epsilon^2}{\delta^2(1-p_0)^2}, &\text{(4.28c)}\\
2\ell^2 + 3\ell + 1 - \frac{6\epsilon^2}{\delta^2(1-p_0)^2} &\geq 0, &\text{(4.28d)}\\
\ell &\geq \frac{\sqrt{1 + \frac{48\epsilon^2}{\delta^2(1-p_0)^2}} - 3}{4}, &\text{(4.28e)}\\
\ell^* &\geq \max\left[1, \left\lceil \frac{\sqrt{1 + \frac{48\epsilon^2}{\delta^2(1-p_0)^2}} - 3}{4} \right\rceil\right]. &\text{(4.28f)}
\end{aligned}$$

In (4.28e), we have kept the root of the quadratic equation that can be positive. In (4.28f), we have ensured that at least one sweep is needed for the test, because the root ℓ in (4.28e) can still be negative for large δ. Finally, we have taken the ceiling function on ℓ because the number of sweeps is an integer. This concludes the proof. □

The proposition allows us to set the required number of sweeps such that the adversary fails Test 4 with overwhelming probability, even if he correctly guesses the start time of the motion.

Table 4.1: Summary of abilities of various adversaries against various RSS authenticator Tests of SFIRE.

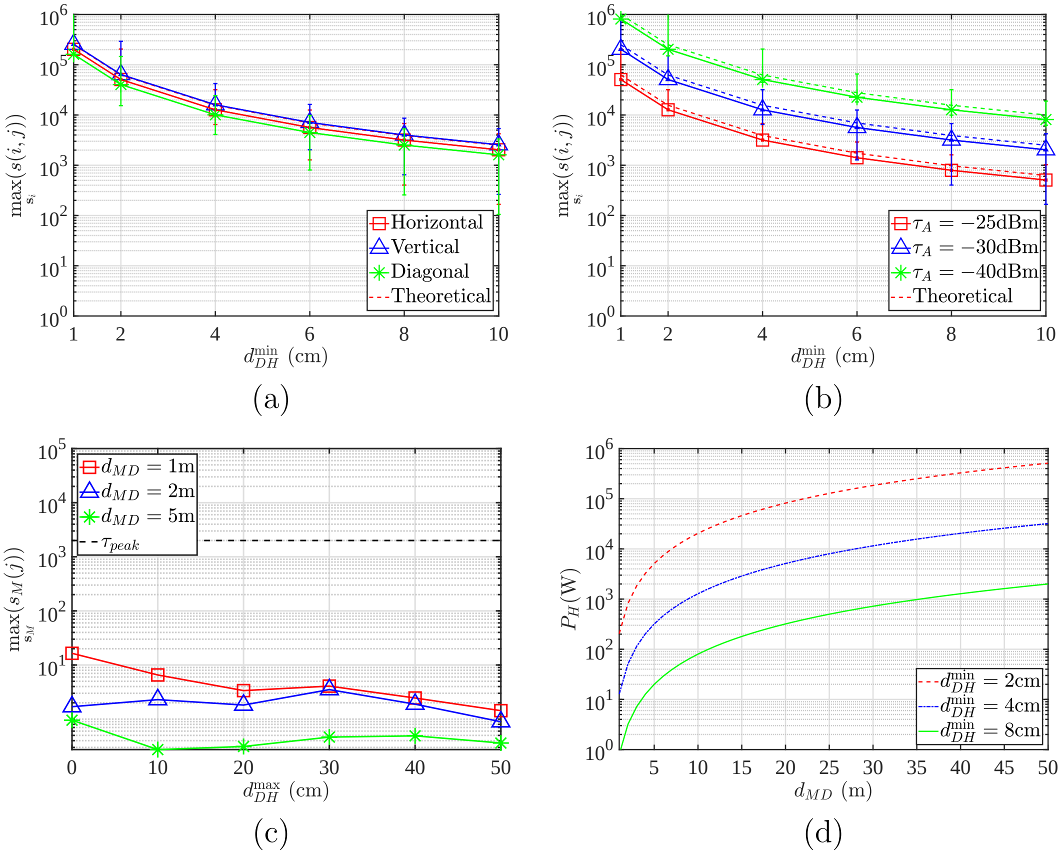
Adversary	Type 1	Type 2	Type 3	Requirement
Test 1	Fail	Pass	Pass	RSS data at H & A
Test 2	Fail	Fail	Pass	RSS data at H & A
Test 3	Fail	Fail	Might Pass	RSS data at H & A
Test 4	Fail	Fail	Fail	RSS data at H & A, accelerometer at H

Table 4.1 summarizes the success of each test against each adversary type and the data requirement for each test.

4.5 Experimental Evaluation

In this section, we experimentally evaluate the security of the RSS authenticator and validate our theoretical analysis. We used two setups in our evaluation. In setup 1, we implemented the RSS authenticator in COTS devices to verify correctness, whereas in setup 2 we used USRP devices to implement the different attacker types and verify soundness. We describe each in detail.

Setup 1–SFIRE with COTS devices: In Setup 1, a Lenovo Y-480 IdeaPad laptop and a Dell XPS desktop, equipped with Intel® Centrino® Wireless N-200 wireless cards were used to implement D and A, respectively. Both cards transmit at 20dBm. The helper H was implemented on a Samsung Galaxy S6 edge+ running Android 7.0 smartphone equipped with an 802.11 a/b/g/n/ac 2.4G+5GHz compatible chipset. The clocks of A and H were synchronized via an Internet server. During the pairing of D with A, we manually performed the three sweeping motions shown in Figure 4.3. A sweeping motion was characterized by three parameters: (a) the minimum distance ($d_{DH}^{\min}$) from D to H, (b) the sweep orientation, and (c) the maximum distance $d_{DH}^{\max}$ from D to H. Minimum and maximum separations were adhered by placing markers on top of D and at the two ends of the motion, although such markers are not necessary for a real protocol execution. During the sweeping



(a)

(b)

(c)

(d)

Figure 4.9: (a) Peak RSS ratio as a function of the minimum D-to-H distances in each sweep for various sweeping motions, (b) peak RSS ratio as a function of the minimum D-to-H distances in each sweep for various RSS at A, (c) peak RSS ratio as a function of the maximum D-to-H distances in each sweep for various d_{MD} for a Type 1 adversary, and (d) maximum transmit power of a Type 2 adversary to achieve τ_{peak} as a function of d_{MD} for various $d_{DH}^{\min}$.

motions, We sampled the RSS at a rate of 10 samples/sec at both H and A and repeated each sweeping motion 1,000 times (35 min approximately).

Setup 2–SFIRE on USRPs: Setup 2 was used to implement the attacks carried out by M on the RSS authenticator tests. The roles of D, A, and M were implemented by three NI-USRP 2921 radios operating at 2.4GHz. The helper radio had a smartphone attached to the top to collect accelerometer data for Test 4. The clocks of all the entities were synchronized via the same computer.

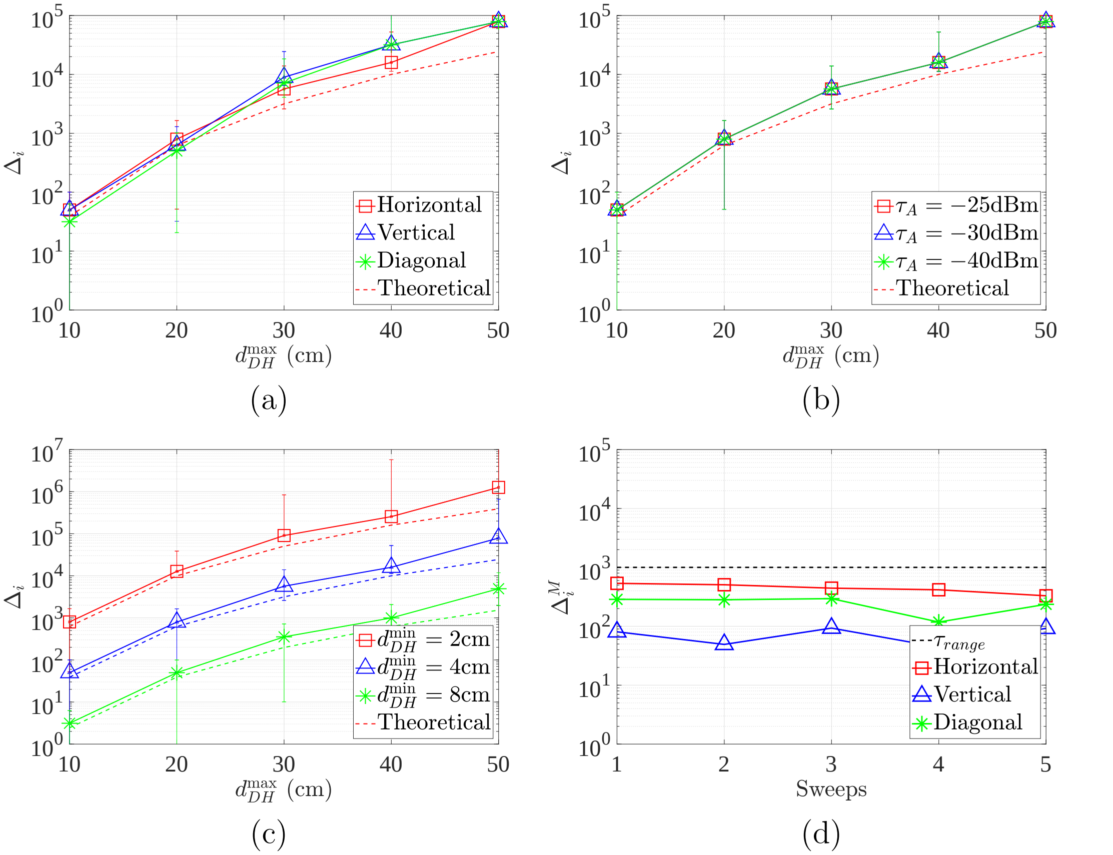
4.5.1 Test 1: Peak RSS Ratio

To evaluate the peak RSS ratio $\max_{\mathbf{s}_i} s(i,j)$ achieved during a benign scenario, two experiments were performed using Setup 1. In the first experiment, D was placed at 5m from A such that the average RSS at A was -40dBm and H was swept over D. In Figure 4.9(a), we show the peak RSS ratio as a function of $d_{DH}^{\min}$ for all the sweeping motions. We observe that the peak RSS obtains very similar values, irrespective of the motion orientation. These values exceed 10^3 for all minimum separations. The theoretical values computed from eq. (4.2) are also shown and match the experimental ones.

In the second experiment, we varied the distance d_{DA}, such that the RSS at A also varied. In Figure 4.9(b), we show the peak RSS as a function of $d_{DH}^{\min}$. The theoretical values computed from eq. (4.8) are also shown. As expected, the peak RSS decreases as D gets closer to A (higher RSS at A), but still maintains large values. This is because the RSS is primarily dominated by $d_{DH}^{\min}$. The plots in Figure 4.9(a) and 4.9(b) can be used to select the threshold τ_{peak} for Test 1.

Detecting a Type 1 adversary: To demonstrate the detection of a Type 1 adversary, we performed an experiment using Setup 2. We fixed the D-to-A distance to 5m and chose the corresponding threshold as $\tau_{peak} = 2{,}000$, based on Figure 4.9(b). We measured the peak RSS ratio when the adversary M was placed at $d_{MD} = 1$m, 2m, and 5m from D and H, respectively. The adversary was set to transmit at 1W. Figure 4.9(c) shows the peak RSS ratio achieved by the Type 1 adversary for different values of the maximum distance between D and H for the horizontal motion. We observe that the peak RSS ratio achieved by the transmission of M is significantly lower than the threshold τ_{peak}. This is because a Type 1 adversary transmitted using an omnidirectional antenna affecting the received power both at H and A thus maintaining a relatively low ratio.

Defeating Test 1 with a Type 2 adversary: Now, we evaluate the transmit power required by a Type 2 adversary to defeat Test 1. We compute the transmit



(a)

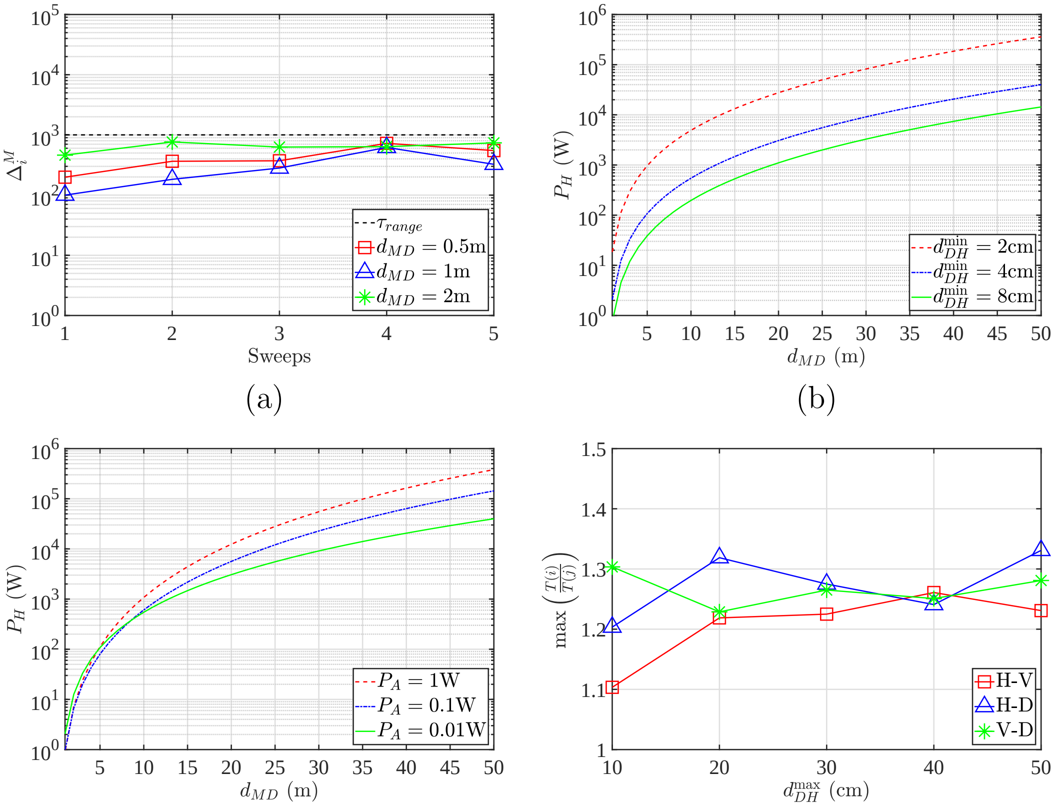
(b)

(c)

(d)

Figure 4.10: (a) RSS ratio range of $\mathbf{s}_i$ as a function of the maximum D-to-H distances in each sweep for various sweeping motions, (b) RSS ratio range of $\mathbf{s}_i$ as a function of the maximum D-to-H distances in each sweep for various RSS at A, (c) RSS ratio range of $\mathbf{s}_i$ as a function of the maximum D-to-H distances in each sweep for various $d_{DH}^{\min}$, and (d) RSS ratio range as a function of the sweeps for a Type 2 adversary for $d_{MD} = 2$m with τ_{range} selected for $d_{DH}^{\max} = 50$cm and $P_A = 0.1$W.

power according to (4.11). We set the threshold for Test 1 to $\tau_{peak} = 2{,}000$ according to the previous experiments using Setup 1. We set the values for all the antenna gains to one. We set the signal strength required to perform an overshadowing attack at A to -50dBm, In Figure 4.9(d), we show the required transmit power of a Type 2 adversary to defeat Test 1 as a function of the device-to-adversary distance (d_{MD} in meters). We consider various minimum D-to-H distances during the sweep. We observe that the required transmit power that satisfies the peak RSS ratio and



(a)

(b)

(c)

(d)

Figure 4.11: (a) RSS ratio range as a function of the sweeps for Type 2 adversary with τ_{range} selected for $d_{DH}^{\max} = 50$cm and $P_A = 0.1$W, (b) maximum transmit power of a Type 3 adversary to H as a function of d_{MD} when transmitting with $P'_A = 10$W for achieving τ_{range} for various $d_{DH}^{\min}$, (c) maximum transmit power of a Type 3 adversary to H as a function of d_{MD} when transmitting with $P'_A = 10$W for achieving τ_{range} for various τ_A, and (d) $\max\left({}^{T(i)}/{}_{T(j)}\right)$ as a function of the maximum D-to-H distances for each motion for various RSS at A.

achieves an overshadowing attack becomes prohibitive with the increase of the D-to-M distance. At 10m from D, the adversary must transmit at hundreds of watts to the helper for achieving the required ratio.

4.5.2 Test 2: RSS Ratio Dynamic Range

We performed three experiments to evaluate the dynamic range Δ_i for all sweeping motions using Setup 1. In the first experiment, we placed A at 10m from D so that the average received RSS at A was -40dBm, Moreover, we fixed $d_{DH}^{\min} = 4$cm and performed horizontal, vertical, and diagonal sweeping motions. For each motion we recorded the dynamic RSS ratio range. Figure 4.10(a) shows the RSS ratio range as a function of the maximum separation between D and H. The theoretical values computed using eq. (4.12) are also shown. In the second experiment, we varied the distance between A and D and repeated the measurements. Figure 4.10(b) shows the RSS ratio range for the different RSS thresholds at A. For both experiments, it can be observed that the range does not vary significantly with the motion orientation. Moreover, the theoretical values match track the measured values. The recorded differences are due to the free-space model considered in the theoretical calculation, however, they can serve as a lower bound on the expected Δ. Longer sweeps significantly increase the RSS ratio range. Figure 4.10(c) shows the results of our third experiment where we varied $d_{DH}^{\min}$ for a horizontal sweeping motion. As expected, the maximum range is achieved when H is swept at 2cm from D and the range of H's motion is maximized (50cm). Based on these results, we set $\tau_{range} = 10^3$ which captures any motion over 30cm with $d_{DH}^{\min} = 4$cm.

Detecting a Type 2 adversary: We now evaluate the ability of a Type 2 adversary in defeating Test 2 using Setup 2. We equipped two USRP devices with directional antennas pointing to H and A, respectively. One USRP antenna was pointed to the hub and transmitted at $P_A = 0.01$W, the minimum required value to successfully perform an overshadowing attack. The other USRP antenna was pointed to the helper and transmitted at $P_H = 1$W to achieve an overshadow attack but also achieve the maximum RSS ratio threshold required in Test 1. Figure 4.10(d) shows Δ_i^M achieved by the adversary for various motions when the distance between H and M is as low as 2m. The adversary's RSS ratio range is below τ_{range} for most motions and reaches the required range only for one horizontal sweep. The adversary failed Test 2, as it needed to pass the test for all sweeps. The horizontal motion exhibited the highest RSS ratio range because we positioned M at the optimal position L_M^* shown in Figure 4.4. However, other motions failed to achieve a similar range. We further repeated our experiments for multiple sweeps and for different distances between M and D. The results in Figure 4.11(a) show that even if M is very close to D (within 0.5m), it cannot achieve the required dynamic range consistently, without employing power control.

Defeating Test 2 with a Type 3 adversary: We further calculated the required transmit powers of M to defeat Test 2 according to the conditions of (4.19). A Type 3 adversary varies the transmission power to the helper between P'_H and P_H, and to the hub between P_A and P'_A. The strategy of M for achieving τ_{range} is to maximize the ratios ${}^{P'_A}/_{P_A}$ and ${}^{P_H}/_{P'_H}$. The minimum transmission powers P'_H and P_A of M are governed by (4.6), which expresses the power required for overshadowing. We set these to $P'_H = 0.1$W and $P_A = 0.1$W corresponding to fixed D-to-H and D-to-A distances. According to (4.18), P'_A and P_H have the same effect on Δ_M. To see the trend of P_H, we fix the value of $P'_A = 10$W. In Figure 4.11(b), we plot the maximum transmit power to H as a function of the D-to-M distances for different minimum D-to-H distances 2cm, 4cm, and 8cm.

We also varied the minimum transmit power to A to values $P_A = 1$W, $P_A = 0.1$W, and $P_A = 0.01$W, while keeping $P'_H = 0.1$W constant (varying P'_H has the same effect on Δ_M). Figure 4.11(c) shows the required maximum transmit power for a Type 3 adversary as a function of d_{MD} for defeating Test 2. From Figure 4.11(b) and Figure 4.11(c) we observe that the required transmit power becomes quickly prohibitive as the adversary moves further away. At 10m from D, the adversary must transmit at hundreds of watts to achieve the required dynamic range. It should be noted here, if the adversary fixes P_H, the variation of P'_A follows similar patterns. The adversary may be able to achieve the required peak ratio if he employs highly directional antennas and manages to be in close distance to H during the pairing.

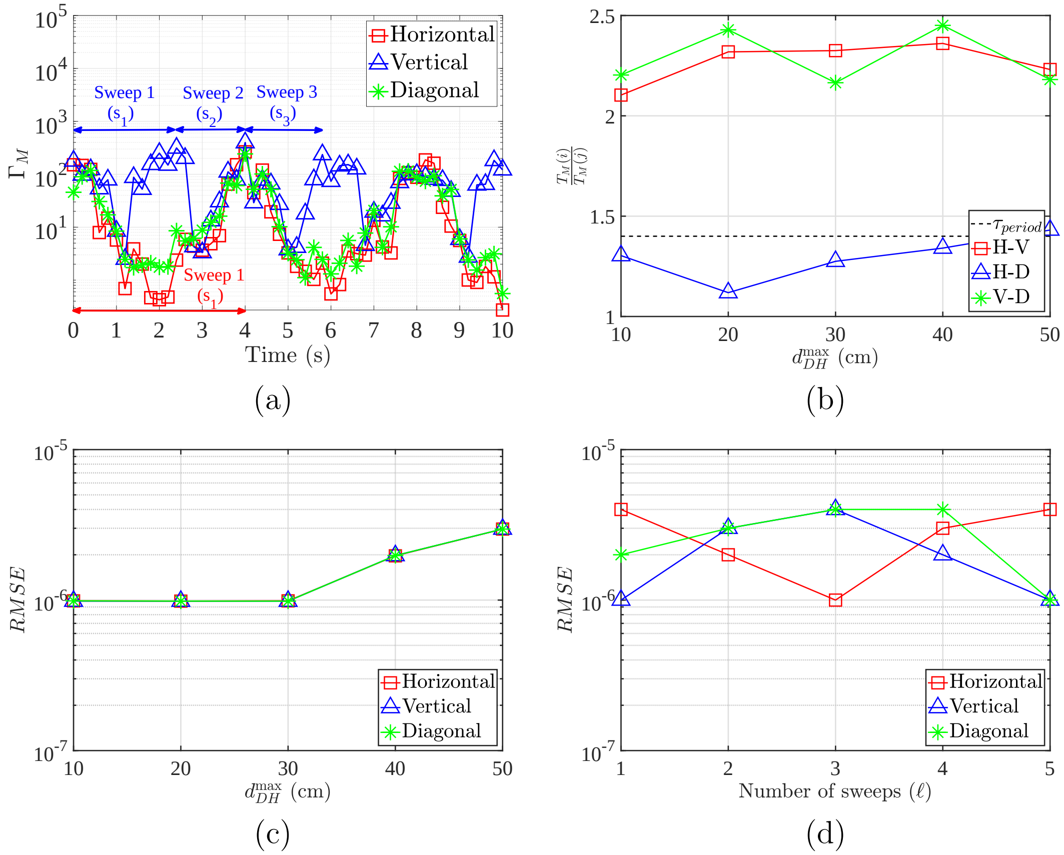


Figure 4.12: (a) RSS ratio fluctuation for a Type 2 adversary as function of the time at 2m from D, (b) $^{T_M(i)}/_{T_M(j)}$ as a function of the maximum D-to-H distances for a Type 3 adversary mimicking transmit power for vertical sweeping motion of H, (c) $RMSE$ as a function of the maximum D-to-H distances for various sweeping motions, and (d) $RMSE$ as a function of the number of sweeps (ℓ) for various sweeping motions.

4.5.3 Test 3: Sweep Period

For Test 3, we performed two experiments using Setup 1 to evaluate the consistency of the sweeping periods across different motion orientations. In the first experiment, we moved the helper on top of the device D and measured the ratio of the sweep periods between pairs of motions; horizontal-vertical (H-V), horizontal-diagonal (H-D) and vertical-diagonal (V-D). Figure 4.11(d) shows the period ratio for all the motion combinations as a function of $d_{DH}^{\max}$. We observe that the sweep period is relatively constant with period ratios not exceeding 1.32. Moreover, the periods did not vary much with the motion range. Based on these experiments, we set $\tau_{period} = 1.4$.

Detecting a Type 3 adversary: Since a Type 3 adversary is the only model tht can defeats Tests 1 and 2, we evaluated if a Type 3 adversary can defeat Test 3. We considered that M is aware of the average period of H's sweeps and regulated its power control accordingly. We employed Setup 2 to allow for power control and antenna directionality, fixed the distance between $d_{MD} = 1$m, $d_{DH}^{\max} = 50$cm, and $d_{DH}^{\min} = 4$cm. M oscillated its transmitting power between 0.01W and 1W to meet both the τ_{peak} and τ_{range} thresholds and defeat Tests 1 and 2. For the experiments, M attempted to synchronize with D's transmission for the vertical motion, with an average period of 2sec, corresponding to an average hand moving speed of 0.5m/s. The user randomized the motion direction. In Figure 4.12(a), we show the RSS ratio fluctuation achieved by the power-controlled transmission of M over time. It can be observed that the sweep period of the vertical sweep is around 2 sec, but the periods of other sweeps are twice as long because only one peak occurs on every sweep (when H is closest to M). Figure 4.12(b) shows the sweep period ratios for different $d_{DH}^{\max}$. When the vertical motion is compared to other motions, the sweep period ratio is over 2. The adversary can pass this test only when the user restricts the helper motion to one orientation (vertical motion in our experiments).

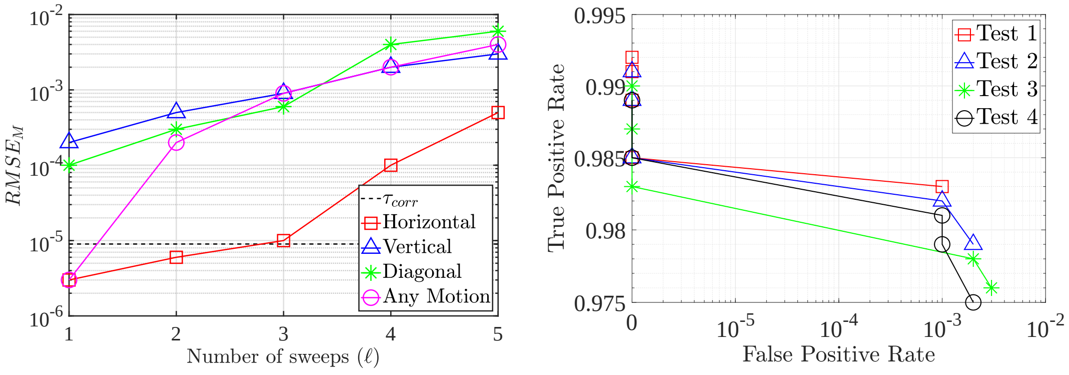
4.5.4 Test 4: RSS Ratio and Motion Correlation

To remedy a possible failure of Test 3 due to using just one orientation, we further considered the correlation of the accelerometer data with the RSS ratio data as dictated by Test 4. We used Setup 1 to evaluate the root mean square error ($RMSE$) between the set of time instances $\mathbf{t}_{RSS}$ when the RSS ratio minimum is measured and the time instances $\mathbf{t}_{acc}$ when an acceleration peak is achieved. The acceleration values were recorded by accessing the accelerometer data on the mobile phone (helper). Figure 4.12(c) shows the average $RMSE$ as a function of the maximum D-to-H distance for various sweeping motions. We observe that the RMSE is quite small indicating the the RSS ratio valleys and acceleration peaks remain synchronized throughout the different motions. Figure 4.12(d) shows the $RMSE$ as a function of number of sweeps (ℓ) for various sweeping motions. We observe no particular correlation between the number of sweeps and the RMSE. This is consistent of our intuition for a benign scenario where the RMSE is not cumulative with the number of sweeps, but it rather varies in a random fashion. Based on the results of this experiment, we set $\tau_{corr} = 9 \times 10^{-5}$.

Detecting a Type 3 adversary: We considered a Type 3 adversary attempting to defeat Test 4 by employing Setup 2. In this experiment, the adversary applied power control and attempted to synchronize to the user motion. We evaluated the best case scenario for the adversary where he had knowledge of motion start time ($\mathbf{E}_M = 0$) and of the average sweep period, which was $T = 2$sec. The synchronization of the adversary's power control with the helper's motion was performed offline by offsetting the first RSS ratio minima to match the first acceleration maxima. Figure 4.13(a) shows the achieved $RMSE_M$ for various sweeping motions of H as a function of the number of sweeps, when the adversary mimicked the horizontal sweep motion. We observe that the error induced in the sweep period when the user moves M accumulates with ℓ leading to the eventual failure of Test 4. Moreover, when the motion mimicked by the adversary is different than that performed by H, even one sweep is sufficient to lead to high RMSE values.

4.6 Evaluation of SFIRE-enabled Device Pairing

We now analyze the security of the device pairing protocol proposed in Section 4.3.2. We first examine if the adversary can pair a rogue device with A. We then examine if D can be deceived to pair with a rogue hub.



(a) (b)

Figure 4.13: (a) $RMSE_M$ as a function of the number of sweeps for a Type 3 adversary, mimicking a horizontal sweeping motion, and (b) ROC curve for the performance of verification tests of SFIRE against various types of adversaries.

4.6.1 Pairing a Rogue Device with A

The pairing of a rogue device D' with A can occur under two different scenarios:

Pairing in the absence of a legitimate device: The pairing protocol described in Section 4.3.2 is initiated with the press of a button on H and D. The button pressing sends a pairing initialization message to the A which is authenticated using the secure AE($\cdot$) function. Without access to the helper, the adversary cannot initiate the pairing from a remote location.

Hijacking a legitimate pairing session: Since M cannot initiate the pairing process with the A, he can only attempt to pair a rogue device with the A by hijacking a pairing session involving a legitimate device (D). To establish a secret key with the A, the adversary must modify the DH public number z_D of D into its own DH public number z'_D, where z_D is contained in the first message m_D sent from D to the A (similar to a typical MitM attack against a DH key exchange). However, m_D is protected by our integrity verification primitive of SFIRE.

As discussed in this Section earlier, the adversaries with different capabilities are not able to pass the RSS authentication to forge m_D. Therefore, the adversary will be unable to pair D' with the legitimate A.

4.6.2 Pairing D with a Rogue Base Station

We now examine if M acting as a rogue A can pair with D. To do so, M can perform a similar MitM attack as in the uplink direction, by replacing the A's DH public parameter z_A with its own z'_A. The m_A is protected by downlink SFIRE primitive $[\cdot]_{Dw}$ as discussed in Section 4.3.3. In the downlink SFIRE, D computes Γ, during the transmission of m_A from RSS values of frames received from A and H. D performs the RSS authentication that prevents pairing with A'.

4.6.3 ROC Curves

We evaluated the receiver operating characteristic (ROC) curve for the SFIRE-enabled pairing protocol. We evaluated the performance of each adversary types against the four tests on Setup 2. The distance between M and D was set to 1m. The value for τ_{peak} was chosen as 2,000 for $P_A = 0.1$W, $\tau_{range} = 10^3$ for the same transmit power to A, $\tau_{period} = 1.4$ and $\tau_{corr} = 9 \times 10^{-6}$. The sweeping motions for each experiment were repeated $1,000$ times. The D to A and D to M distance were fixed to 1m and 0.5m respectively, with M positioned at L^*_M as shown in Figure 4.4.

Figure 4.13(b) shows the ROC curve for all the tests in the RSS authenticator. Test 1 is evaluated against a Type 1 adversary, Test 2 is evaluated against a Type 2 adversary, and Tests 3 and 4 are evaluated against a Type 3 adversary. The various points of the ROC curve are obtained for a different number of sweeps to complete the protocol. The rightmost point is obtained for one sweep, whereas the leftmost point is obtained for five sweeps. We observe that as the number of sweeps increases, the TPR increases whereas the FPR decreases indicating that the SFIRE protocols achieve both correctness and security.

4.6.4 Timing Performance

The timing performance is dictated by the time to complete the number of sweeps. This time dominates the computation of DH key, transmission delays, etc. From Figure 4.13(b), three sweeps give zero false positive rates for all Types of M. The time to complete the protocol requires six sweeps (three for uplink and three for downlink), translating to 6sec.

4.7 Chapter Summary

We addressed the problem of secure device pairing without prior associations. We proposed SFIRE, a secret-free protocol that achieves the secure pairing of COTS wireless devices with a hub. Compared to the state-of-the-art, SFIRE does not require any out-of-band channels, special hardware, or firmware modification, thus it is applicable to any COTS device. We showed that SFIRE is resistant to the most advanced active signal manipulations that include recently demonstrated signal nullification at an intended receiver. These security properties are achieved in-band with the assistance of a helper device and by using the RSS fluctuation patterns to build a robust RSS authenticator. We performed extensive theoretical analysis and attested the finding with experiments using COTS devices and USRP radios and validated the security and performance of the proposed protocol.

CHAPTER 5

SECURE DEVICE BOOTSTRAPPING WITHOUT SECRETS RESISTANT TO SIGNAL MANIPULATION ATTACKS

5.1 Introduction

5.1.1 Motivation

In a number of scenarios, such as a body area network, home IoT, a battlefield ad-hoc network, etc. a group of devices as shown in Figure 5.1 may be required to bootstrap in a short amount of time. HELP presented in Chapter 3, SFIRE presented in Chapter 4, and other secure device pairing methods [20–32, 34, 35, 92] do not scale with the number of devices. In a group setting, the user would have to be manually execute the protocol at multiple locations and device pairing must occur sequentially. There are two major issues with such extensions. First, the user effort becomes significant with OOB channel pairing, if it has to be repeated

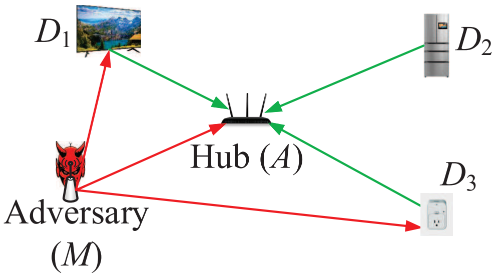


Figure 5.1: Multiple devices D_1, D_2, and D_3 bootstrapping with the hub (A) in presence of an MitM adversary (M).

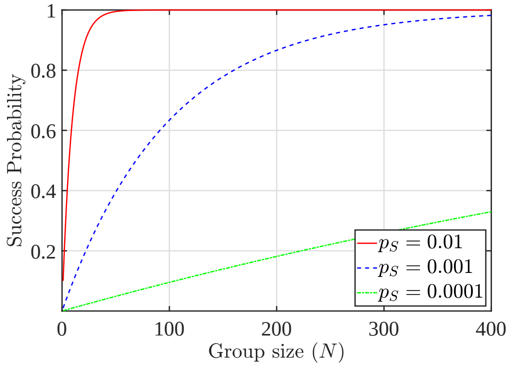


Figure 5.2: System model depicting all entities.

multiple times. Second, as it was shown by Mirzadeh *et al.* [107], suppose the success probability of an adversary pairing with the system to be p_S. With N pairing repetitions, the adversary's success probability of pairing one device becomes $1-(1-p_S)^N$ which approaches one with N, graphically shown in Figure 5.2. In our work VERSE, we leverage the existence of multiple devices to actually reduce the probability of a successful attack. Moreover, orthogonal to these works, our method requires the least user interaction (powering of devices and initialization of pairing from the hub). The message integrity verification is done in-band for all the participating devices without requiring any other interface (led lighting, microphone, speaker) other than the common RF interface. Also, most prior works do not address the possibility of MitM attacks, where the adversary can hijack the session of a legitimate entity by performing signal cancellation and injection. Compared to HELP, the only other work that addresses an MitM over wireless without pre-shared secrets, VERSE does not require a helper with an authenticated channel to the hub that also needs to be manually moved by the user. Moreover, VERSE improves security with a group of devices. Finally, the security of VERSE does not hinge on the close proximity of some devices, the randomness of the channel, nor the placement of the adversary outside a protected zone. Rather, it is derived from the fundamental constraints posed by the geometry and basic signal propagation properties.

In this chapter, we address the problem of securely bootstrapping multiple devices with a single entity such as a hub or a base station. Our goal is not to differentiate between legitimate and malicious devices. Such a proposition is infeasible in the absence of any prior trust and without the existence of out-of-band channels for verification, or some unique advantage of the legitimate devices (proximity, superior channel conditions, unique contextual information, etc.). Rather, we aim to guarantee protocol soundness in the absence of an adversary, and abort the bootstrapping process if any active protocol manipulation is detected. Moreover, we investigate if the presence of multiple legitimate devices can be leveraged to strengthen resistance to signal cancellation and therefore improve the security of the pairing process. We theoretically and experimentally characterize the limits of the adversary's capability based on geometric constraints and exploit those limits to construct a secure bootstrapping protocol for multiple devices.

5.1.2 Main Contributions and Chapter Organization

Our main contributions are four-fold:

- We develop a scalable PHY-layer group message integrity verification primitive called VERSE that achieves bootstrapping in-band (using only a common RF interface) and does not rely on pre-shared secrets. The key idea is to simultaneously verify the integrity of a transmitted message at multiple receivers, thus forcing the adversary to perform signal cancellation/injection at multiple locations simultaneously. This requirement dramatically degrades the success of MitM over wireless.

- We use VERSE to construct a secure in-band bootstrapping protocol for multiple devices based on the Diffie-Hellman (DH) key agreement. Our protocol securely pairs and then establishes pairwise keys with the hub. Such keys can then be used to establish group keys, if necessary.

- We analyze the security of VERSE and theoretically establish that a successful

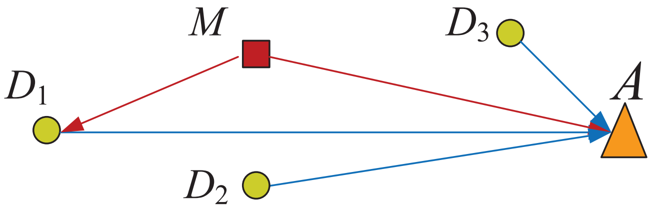


Figure 5.3: System model depicting all entities.

attack becomes infeasible if three or more verifiers are present when a single malicious device launches the attack. Moreover, the effort of a multi-device adversary must scale linearly with the group size.

- We carry out extensive USRP experiments to evaluate the effectiveness of our PHY-layer integrity verification against signal manipulations. First, we demonstrate the effectiveness of cancellation and injection attacks over a single channel. We then evaluate signal manipulations when multiple devices are used as receivers and/or transmitters and validate our theoretical findings. We then evaluate the adversary's ability to defeat VERSE.

Chapter Organization: In Section 5.2, we describe the system and adversary models. We present the VERSE primitive and the secure bootstrapping protocol for multiple devices in Section 5.3. We analyze the protocol's security in Section 5.4. The experimental evaluation of MitM attacks over wireless and of the security of our protocol are detailed in Section 5.5. We conclude the chapter in Section 5.6.

5.2 Model Assumptions

5.2.1 System Model

We consider the system model shown in Figure 5.3. The system consists of the following entities:

Hub (A): The hub coordinates and verifies the bootstrapping process. It is assumed to be under user control.

Legitimate Devices (D): We consider a set of legitimate devices $\mathbf{D} = \{D_1, D_2, \ldots, D_{N-1}\}$ that are newly introduced into the network. The devices attempt to pair with A, but do not share any prior secrets. They are assumed to be under user control. The devices and A are synchronized to a common slotted system with a bounded synchronization error ϵ. Synchronization is achieved with any known method such as [90], and it is already a necessary requirement for many standardized MAC protocols that follow a time slotted system [108–111].

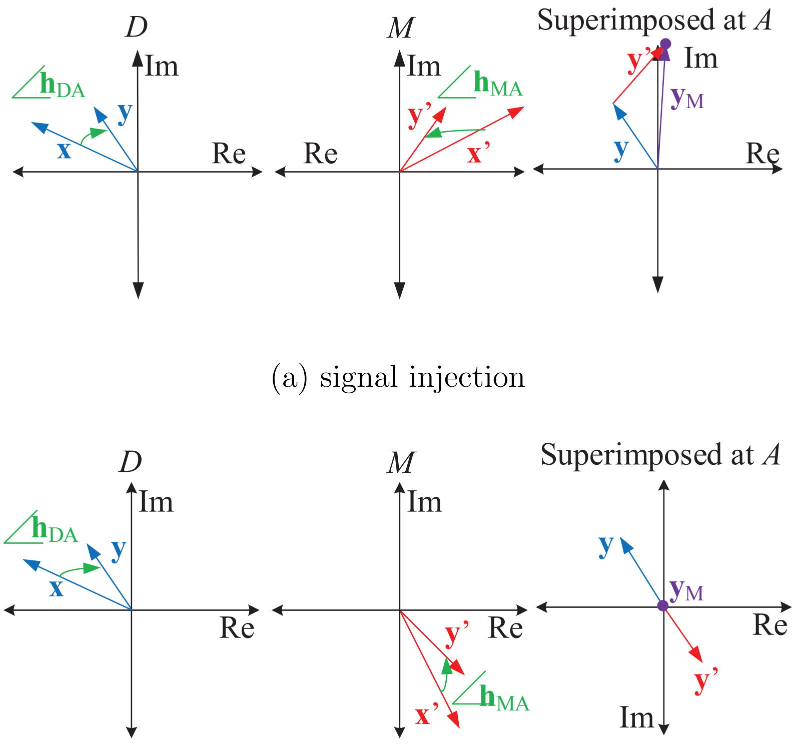
Adversary (M): We consider an active adversary that aims at (a) pairing with A as a legitimate device and (b) spoofing a rogue hub that is joined by at least one legitimate device. We do not address DoS attacks such as jamming, simply aiming at preventing the pairing of legitimate devices without gaining access to the system. All entities are located within the same collision domain and can overhear broadcast transmissions.

5.2.2 Threat Model

We consider an adversary that is aware of the protocol executed by the legitimate parties but does not have physical access to any of the devices. Because the bootstrapping process is initiated by the user, the adversary can only hijack an ongoing session. This can be achieved by launching an MitM attack and modifying the wireless transmissions during the bootstrapping session. We analyze the feasibility of the MitM attack when the adversary deploys a single device. We further discuss the feasibility and complexity of a multi-device MitM attack.

MitM attack by a single device

Let a legitimate device D transmit a message m to the hub A. To perform an MitM attack, the adversary has to replace m with m'. Let $\mathbf{x} = \{x(1), x(2), \ldots, x(k)\}$



(a) signal injection

(b) signal cancellation

Figure 5.4: (a) A signal injection attack and (b) a signal cancellation attack.

denote the transmitted symbols modulating m and $\mathbf{y} = \{y(1), y(2), \ldots, y(k)\}$ the received symbols at A. Then,

$$\mathbf{y} = \mathbf{h}_{DA}\mathbf{x}, \tag{5.1}$$

where $\mathbf{h}_{DA} = \alpha_{DA} \cdot e^{j\phi_{DA}}$ is the impulse response of the D-A channel, α_{DA} is the channel attenuation factor, and ϕ_{DA} is the channel's phase shift. Here, we have assumed that the entire transmission of $\mathbf{x}$ completes within the channel's coherence time, so the channel remains constant. To modify $\mathbf{y}$, the adversary M must transmit $\mathbf{x}'$, modified by the M-A channel to $\mathbf{y}' = \mathbf{h}_{MA}\mathbf{x}'$ such that the superposition $\mathbf{y}_M =$

$\mathbf{y}+\mathbf{y}'$ decodes to m'. In other words, M must compute

$$\mathbf{x}' = \frac{1}{\mathbf{h}_{MA}}(\mathbf{y}_M - \mathbf{h}_{DA}\mathbf{x}), \tag{5.2}$$

and transmit $\mathbf{x}'$ in a timely fashion such that $\mathbf{y}$ and $\mathbf{y}'$ are superimposed as shown in Figure 5.4(a). According to equation (5.2), the computation of $\mathbf{x}'$ requires the knowledge of the signal $\mathbf{x}$ transmitted by D and of the channels $\mathbf{h}_{DA}$ and $\mathbf{h}_{MA}$. Moreover, the reception of $\mathbf{y}'$ must be synchronized with the reception of $\mathbf{y}$ such that $\mathbf{y}'$ arrives at A within an acceptable delay spread τ_A for correct symbol superposition [89]. Synchronization can be achieved using the preambles or the pilot symbols from the device; such methods are discussed in detail in [90]. The delay spread requirement imposes an important physical constraint on M's locations. The difference between the adversary's path, and the direct path must satisfy

$$d_{DM} + d_{MA} - d_{DA} \leq \tau_A \cdot c, \tag{5.3}$$

where d_{XY} denotes the distance between X and Y and c is the speed of light.

When the signal $\mathbf{x}$ is MC ON-OFF encoded, denoted by $[\mathbf{x}]$, modification of the received signal to $\mathbf{y}_M$ requires some ON slots of $[\mathbf{x}]$ to be annihilated, i.e., the amplitude of $\mathbf{y}_M$ must be below the signal detection threshold (typically 10s of dBms below zero) in some slots. Hence, the adversary must be capable of carrying out a signal cancellation attack. We primarily focus on the cancellation scenario, because it is more challenging to achieve than shifting the original constellation point closer to another point in the I-Q plane. The latter can be achieved by launching an overshadowing attack [54].

Practically, obtaining $\mathbf{x}$ in advance to compute $\mathbf{x}'$ is not possible. This is because D can transmit random symbols to implement an ON slot when ON-OFF keying is used. These symbols do not need to belong to a particular modulation mode such as BPSK, QPSK, etc. Alternatively, the adversary can avoid the requirement of knowing $\mathbf{x}$, by performing a relay attack. In this attack, the adversary's position is strategically selected such that the path difference between the direct path and the adversary's path satisfies:

$$d_{DM} + d_{MA} - d_{DA} = (2w+1)\frac{\lambda}{2}, \quad w = 0, 1, 2, \ldots \tag{5.4}$$

where λ denotes the wavelength. This guarantees that the inverse of $\mathbf{y}$ will be received at A when the incoming signal at M is compensated for the respective channel attenuation factors. Because the path difference is an odd multiple of $^{\lambda}/_{2}$, $\mathbf{y}$ and $\mathbf{y}'$ arrive at A with opposite phases, thus canceling each other ($\mathbf{y}_M = 0$). The signal superposition at A for a cancellation attack is shown in Figure 5.4(b). To enable a fast and error-free relay operation, the adversary may be equipped with directional antennas, one for receiving the transmission of D and one for relaying $\mathbf{x}'$. (5.4) can be generalized for the adversary who is capable of modifying the phase (ϕ_{MA}) of the relayed signal in real time. From a geometric standpoint, modifying the phase of the incoming signal only changes the set of ellipses that yield cancellation. The new set of ellipses must satisfy,

$$d_{DM} + d_{MA} - d_{DA} = (2w+1)\frac{\lambda}{2} + \frac{\phi_{MA}}{\pi}, \quad w = 0, 1, 2, \ldots \tag{5.5}$$

We note that the phase calculations in (5.5) assume a strong Line-of-Sight (LoS) environment between all three entities. This is the best-case scenario for M, as it allows the calculation of a location from where cancellation via relaying becomes possible, without knowing $\mathbf{x}$ and by modeling $\mathbf{h}_{DA}$, since the latter cannot be directly measured. In the general case, $\mathbf{x}$ arrives at A via multiple paths which hardens channel modeling. In our model, we consider this best-case scenario for the attacker, where the channel is predictable with a strong LoS.

When M's placement satisfies (5.4) or (5.5) and assuming stable LoS channels, the symbols traveling over the relay path are copies of the symbols received via the LoS path but shifted by $(2w+1)\pi$ and attenuated differently. Therefore, M does not need to know the transmitted symbols a priori. To compensate for the attenuation

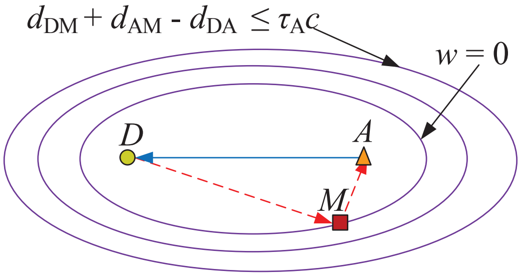

$d_{\mathrm{DM}} + d_{\mathrm{AM}} - d_{\mathrm{DA}} \leq \tau_{\mathrm{A}} c$
$w = 0$
D
A
M

Figure 5.5: To perform signal cancellation, the adversary is placed on an ellipse, centered at D and A that satisfies a path difference of $(2w + 1)\lambda/2$ and does not violate the maximum delay spread τ_A.

difference, M must only know the attenuation factors α_{DA}, α_{DM}, and α_{MA} in the impulse responses $\mathbf{h}_{DA}$, $\mathbf{h}_{DM}$, and $\mathbf{h}_{MA}$, respectively. Some of these channels ($\mathbf{h}_{DM}$, and $\mathbf{h}_{MA}$) can be measured, whereas the $\mathbf{h}_{DA}$ channel can be modeled after a path loss model.

We now examine the candidate set of M's locations that lead to successful cancellation via relaying. The adversary's location ℓ_M must satisfy the phase difference equation in (5.5) and the delay spread constraints in (5.3). For (5.4) or (5.5), candidate ℓ_M form a series of ellipses with D and the A placed at the two focal points. The set of such ellipses is shown in Figure 5.5 and is computed by considering all odd integer values of w in (5.4) or (5.5). Finally, the delay spread constraint (5.3) upper bounds w.

MitM attack by multiple coordinated devices

When the adversary has multiple devices at his disposal, he can deploy them at multiple locations to perform simultaneous signal cancellation at more than one receivers. For instance, each adversarial device may target a single legitimate device. However, this attack requires online coordination among the different devices (timely channel sensing, time synchronization, power coordination, etc.) and the

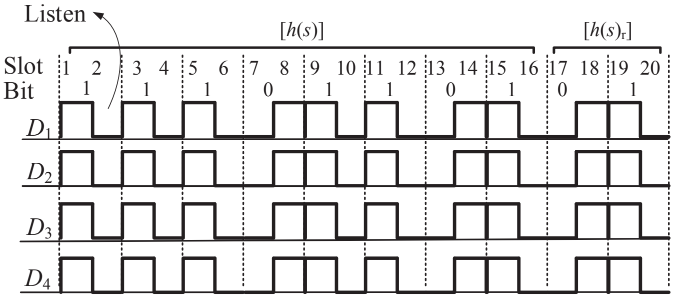


Figure 5.6: D_1, D_2, D_3 and D_4 synchronously transmit $[h(s)]$. The devices sense the channel during the OFF slots.

use of highly-directional transmissions to avoid unintended interference. For IoT scenarios, pairing devices are relatively close, which requires the use of very narrow beams. Antennas that can achieve such narrow beams are bulky with many antenna elements and therefore easily discernible in an IoT environment. Moreover, the attacker's cost increases linearly with the number of legitimate devices that are deployed. We primarily focus on the single device scenario and comment on the security and limitations of our scheme under a multi-device adversary.

5.3 The Secure Bootstrapping Protocol

In this section, we present an in-band secure bootstrapping protocol for a group of devices. We first describe VERSE, a PHY-layer message integrity protection primitive that exploits multiple verifiers to detect signal manipulation attacks launched by an MitM adversary. We then use VERSE to construct an authenticated pairwise key establishment protocol between a group of devices and the hub, based on DH key agreement.

5.3.1 The VERSE Primitive

Consider a general group protocol in which multiple legitimate devices sequentially exchange a set of messages. Let s denote the *protocol transcript*. In VERSE, all legitimate devices operate as verifiers by recording the over-the-air messages. Each device compiles s and contributes in the integrity verification process by broadcasting a transcript digest $h(s)$, where $h(\cdot)$ is a non-cryptographic hash function. Specifically, all verifiers synchronously transmit the MC ON-OFF modulated message $[h(s) \mid\mid h(s)_r]$ where $h(s)_r$ is a repetition of the last r bits of $[h(s)]$. The synchronous transmission $[h(s) \mid\mid h(s)_r]$ is shown in Figure 5.6. During the OFF slots of the $[h(s)]$ transmission, verifiers sense the wireless channel. If any device D_i compiled an $s' \neq s$, there will be at least one OFF slot for which D_i will sense an ON slot, as $h(s') \neq h(s)$ with overwhelming probability. Upon sensing this discrepancy, D_i will raise an alarm by sending only ON slots, essentially jamming the remainder of the $[h(s) \mid\mid h(s)_r]$ transmission, leading to further alarms being raised by the rest of the verifiers. The addition of $h(s)_r$ guarantees that an alarm will be raised, even if an integrity violation is detected at the last bit in $h(s)$.

Formally, the VERSE primitive involves the following steps:

1. **Compilation of the protocol transcript:** Each D_i broadcasts a message m_i using its default modulation mode. These messages are recorded by all D_is. Every D_i compiles the protocol transcript as $s = m_1 || m_2 || \ldots || m_N$.

2. **Device Synchronization:** A lead device (e.g., the hub) sends a delimiter to synchronize the clocks of all D_is. We set the delimiter to be an ON-ON-OFF-OFF-ON-ON sequence, which is not a valid MC-coded sequence.

3. **Transcript digest transmission:** Following synchronization, D_is transmit $[h(s) \mid\mid h(s)_r]$ synchronously using MC ON-OFF keying, where $h(\cdot)$ is a non-cryptographic uniform hash function and $h(s)_r$ are the last r bits of $h(s)$.

4. **Transcript verification:** While $[h(s) \mid\mid h(s)_r]$ is being transmitted, each D_i

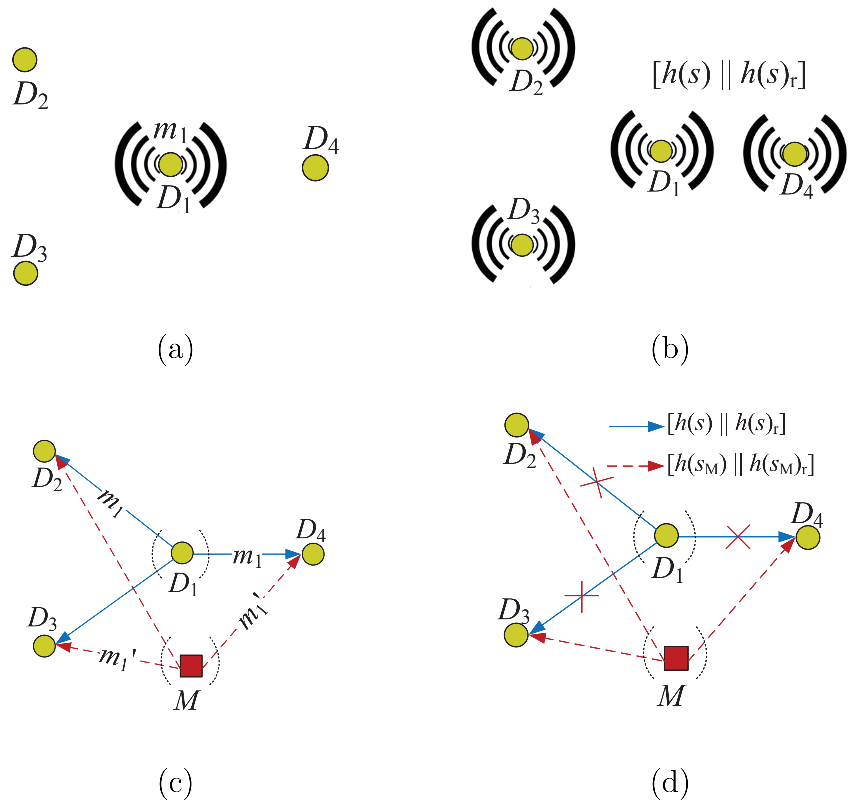


Figure 5.7: (a) Transmission of m_1, (b) synchronous transmission of $[h(s) \mid\mid h(s)_r]$ during the integrity verification phase, (c) M replaces m_1 with m_1' by launching an overshadowing attack, and (d) M attempts a signal cancellation at D_1, D_2 and D_3 while D_1 transmits $[h(s) \mid\mid h(s)_r]$.

plays the role of a verifier. During the OFF slots of $[h(s) \mid\mid h(s)_r]$ D_is senses the wireless channel. If any OFF slot is sensed as ON by D_i, then D_i raises an alarm by transmitting ON slots for rest of the slots in $[h(s) \mid\mid h(s)_r]$. The $[h(s)]_r$ is appended to $[h(s)]$ to ensure there are sufficient slots to raise an alarm even if a mismatch is detected at the last ON-OFF bit of $[h(s)]$. The minimum value of r is two.

An example of VERSE for four devices is shown in Figure 5.7. Initially, the devices exchange messages sequentially, creating a protocol transcript s. The transmission of m_1 is shown in Figure 5.7(a). In the transcript verification step shown in

Figure 5.7(b), all devices synchronously broadcast $[h(s) \mid\mid h(s)_r]$ and use the OFF slots to verify the integrity of $h(s)$.

We provide a sketch of VERSE's security (a detailed analysis is presented in Section 5.4). To successfully launch an MitM attack against VERSE, the adversary must ensure that no alarm is raised. Consider M modifying the protocol transcript from s to s_M by modifying m_i. In Figure 5.7(c), we show M replacing m_1 with m_1'. Even if M launches an overshadowing attack against all devices and successfully replaces m_i, the device D_i that originated m_i compiles s. Because $s \neq s_M$, it follows with overwhelming probability that $[h(s) \mid\mid h(s)_r] \neq [h(s_M) \mid\mid h(s_M)_r]$, due to the collision resistance property of $h(\cdot)$. In fact, for a uniform hash function, the two hashes will differ in approximately half the bits. For the bits where $[h(s) \mid\mid h(s)_r] \neq [h(s_M) \mid\mid h(s_M)_r]$, D_i transmits (receives) when the rest of the devices are sensing (transmitting). To avoid the detection of s by the devices that compiled s_M, the adversary must perform signal cancellation from one TX to many RXs, which becomes increasingly difficult with the number of RXs. Similarly, to avoid detection of $[h(s_M) \mid\mid h(s_M)_r]$, at D_i, the adversary must perform signal cancellation from many TXs to one RX, which also becomes increasingly difficult with the number of simultaneous TXs.

5.3.2 Secure Bootstrapping using VERSE

To bootstrap a set of new devices with the hub, we execute a DH key exchange [49] for establishing pairwise keys over the public channel and use VERSE to protect the integrity of the protocol execution. The bootstrapping protocol consists of the following steps, which are also outlined in Figure 3.3.

1. **Initialization:** A total of $N-1$ legitimate devices D_1, $D_2, \ldots, D_{N-1}$ participate in the group. The protocol is initialized when the user sets the hub (A) to pairing mode and loads the total number of devices N (including A) to A. For a period τ (e.g., two mins), the hub broadcasts a random MC ON-OFF se-

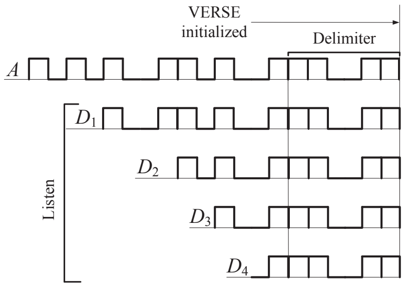


Figure 5.8: Protocol initialization. The hub broadcasts an MC ON-OFF sequence during device activation. This sequence terminates with a known delimiter.

quence that ends in delimiter ON-ON-OFF-OFF-ON-ON. During that period, the user turns on each D_i to set it to pairing mode, and all D_i's synchronize to the MC ON-OFF sequence. Initialization terminates with the delimiter, allowing each device to note the beginning of the DH message exchange phase. Figure 5.8 shows the initialization step for four legitimate devices.

2. **DH message exchange:** All devices use public DH parameters $(\mathbb{G}, q, g)$, where $\mathbb{G}$ is a cyclic group of order q and g is a generator of G. Each D_i broadcasts a message $m_i = ID_i||z_i$ containing ID of D_i and the DH primitive $z_i = g^{X_i}$, where X_i is chosen from $\mathbb{Z}_q$ uniformly at random. The hub also broadcasts $m_A = ID_A||z_A$.

3. **Integrity Verification:** The integrity verification phase is initiated by the transmission of the delimiter by the hub, which serves as a SYNC message for all D_is. The D_is use VERSE to verify the integrity of the protocol transcript $s = m_1||m_2||\ldots||m_{N-1}||m_A$. The hub records the total number of public DH primitives N' exchanged during the protocol execution. The hub verifies that $N \overset{?}{=} N'$ to ensure that the correct number of devices participated in the protocol. If verifications is passed, D_is and A participate in VERSE by transmitting $[h(s) \mid\mid h(s)_r]$. Otherwise, D_is and A raises an alarm by transmitting all ON slots in the remaining of the sequence. The devices stay in pairing mode for a period $\tau' > \tau$ even if the integrity verification is completed. This is to ensure

$D_i \ \forall \ i = 1, \ldots, N-1$		A
Given ID_i,		Given ID_A,
$(\mathbb{G}, q, g)$		$(\mathbb{G}, q, g)$
Pick $X_i \in_U \mathbb{Z}_q$		$X_A \in_U \mathbb{Z}_q$
$z_i \leftarrow g^{X_i}$		$z_A \leftarrow g^{X_A}$
$m_i \leftarrow ID_i, z_i$	$\xrightarrow{m_i}$	$m_A \leftarrow ID_A, z_A$
	$\xleftarrow{m_A}$	
Compute:		Compute:
$s \leftarrow \begin{pmatrix} m_1 \Vert \cdots \Vert \\ m_i \Vert \ldots \Vert \\ m_{N-1} \Vert m_A \end{pmatrix}$		$s' \leftarrow \begin{pmatrix} m_1 \Vert \cdots \Vert \\ m_i \Vert \ldots \Vert \\ m_{N'-1} \Vert m_A \end{pmatrix}$
		Verify; $N \stackrel{?}{=} N'$
	$\xrightarrow{[h(s) \ \Vert \ h(s)_r]}$	
Verify OFF slots in	$\xleftarrow{[h(s) \ \Vert \ h(s)_r]}$	Verify OFF slots in
$[h(s) \ \Vert \ h(s)_r]_i$		$[h(s) \ \Vert \ h(s)_r]_i$
are OFF while sensing.		are OFF while sensing.
Generate,		Generate,
$k_{D_i,A} \leftarrow (z_A)^{X_i}$		$k_{D_i,A} \leftarrow (z_i)^{X_A}$

Figure 5.9: Diffie-Hellman key agreement using VERSE after the initialization step.

that they paired with the legitimate hub and no other pairing operation takes place. If a second MC ON-OFF sequence is overheard by a device D_i, the device raises an alarm.

4. **Confirmation:** Upon successful verification, each device calculates a pairwise key $k_{D_i,A} = g^{X_i \cdot X_A}$. Moreover, A displays a “SUCCESS” message. Else, A displays “FAILURE” and broadcasts a “RESTART” message.

We emphasize that the message integrity verification can be integrated with any group association protocol, such as the group Diffie-Hellman key exchange [112]. For this work, we establish pairwise keys with A. Once pairwise keys are established, A can securely distribute a group key to each device.

5.4 Security Analysis

We first analyze the security of VERSE by demonstrating the infeasibility of signal cancellation when multiple verifiers are used to verify the integrity of the protocol digest. We then evaluate the security of the DH-based protocol presented in Section 5.3.2.

5.4.1 Signal Cancellation from One TX to Multiple RXs

In this section, we analyze the signal cancellation attack for the adversary introduced in Section 5.2. We consider the transmission of an MC ON-OFF sequence from one TX to multiple RXs and show that when at least three RXs act as verifiers, signal cancellation becomes infeasible.

Consider the scenario of Figure 5.11(a), where a transmitter TX broadcasts an MC ON-OFF coded message m_1, which is received by RX$_1$, RX$_2$, and RX$_3$. Let $\mathbf{x}$ denote the symbols of the transmitted message, and $\mathbf{y}_1$, $\mathbf{y}_2$ and $\mathbf{y}_3$ denote the received symbols at RX$_1$, RX$_2$, and RX$_3$, respectively. The ON slots of m_1 are realized by a series of random symbols from the constellation plane, whereas the OFF slots are realized by no transmission. To cancel any ON slot at all three receivers, an adversary M must find a location ℓ_M such that it can simultaneously annihilate $\mathbf{y}_1$, $\mathbf{y}_2$ and $\mathbf{y}_3$, at the respective RXs. This is because $\mathbf{x}$ contains random selected symbols that do not allow the prediction of $\mathbf{y}_1$, $\mathbf{y}_2$ and $\mathbf{y}_3$. Therefore, M must perform a relay attack by being positioned at a location that cancels the received signal at each RX, independently of $\mathbf{x}$.

Let M transmit $\mathbf{x}'$ and RX$_1$, RX$_2$, and RX$_3$ receive $\mathbf{y}'_1$, $\mathbf{y}'_2$, and $\mathbf{y}'_3$. The cancellation attack is successful if $\mathbf{y}'_1 = -\mathbf{y}_1$, $\mathbf{y}'_2 = -\mathbf{y}_2$ and $\mathbf{y}'_3 = -\mathbf{y}_3$. That is, M's transmission arrives at each RX location with an inverse phase and the same amplitude as $\mathbf{y}_1$, $\mathbf{y}_2$ and $\mathbf{y}_3$. For each RX, M's location must satisfy the phase difference equation (5.4). The solution to (5.4) is an ellipse with TX and RX located at the focal points. For three RXs, ℓ_M must lie in the intersection of three ellipses, as

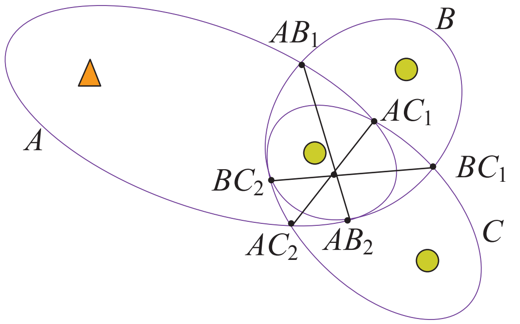


Figure 5.10: Three eclipses sharing one focus point. The lines join the intersection points between pairs of ellipses are concurrent, with the common intersection point inside all three ellipses.

shown in Figure 5.11(a). However, the following proposition shows that no such location exists.

Proposition 7. *Three distinct ellipses sharing one focal point irrespective of the plane they lie in, do not have a common point of intersection.*

Proof. Let A, B, and C be three ellipses sharing a focal point, with the three ellipses being distinct. Each pair of ellipses will have a minimum of two intersection points. Let AB_1, AB_2, BC_1, BC_2, AC_1, and AC_2 be the respective intersection points between A, B, B, C, and A, C. These points are shown in Figure 5.10. According to Theorems 1 and 2 in [113], the lines connecting the intersection points between each pair of ellipses *are concurrent* at a common intersection that lies inside all three ellipses, irrespective of the planes the ellipses lie in. Assume now that there is a common intersection point between all three ellipses. Without loss of generality, assume that AB_1 is the same as AC_1. Then the lines AB_1-AB_2 and AC_1-AC_2 will have a common origin point. The only way that the two lines AB_1-AB_2 and AC_1-AC_2 are concurrent with the BC_1-BC_2 line is if also AB_2 is the same point as AC_2. In the latter case, B and C become the same ellipse or A and B become the same ellipse, and there are no longer three distinct ellipses. Hence, A, B, and C sharing a focal point cannot have a common point of intersection.

The proof states that three ellipses sharing a common focus point cannot have a common intersection point, regardless of the plane that the lie in. This is sufficient for our purposes. Without attempting a formal proof, it is natural to conjecture that the proof does extend to the case of ellipsoids. Ellipsoids consist of an infinite number of ellipses on different planes that have common foci. If three of these ellipsoids share a single focal point, then we can treat their intersection as the intersection of an infinite number of combinations between three ellipses sharing the focal point on different planes. Applying the proof on those ellipses shows that three ellipsoids sharing one focal point do not have a common intersection point. □

Based on Proposition 7, there is no location such that M can perform simultaneous cancellation of the TX's signal at three RXs with a single transmission. There are some degenerate RX arrangements that make cancellation from a single location possible. This is when two of the RXs are at the same location, in which case only the intersection of two ellipses needs to be considered. We consider such cases to be point-specific, which could be avoided by requesting distinct RX locations or including additional verifiers. Moreover, cancellation becomes possible if M is positioned at the common focal point, i.e., at the same location as the TX, which is detectable by the user.

Extending Proposition 7, no common intersection point exists for $n > 3$ if such point cannot be found for $n = 3$. Furthermore, common intersection points between two ellipses exist as shown in Fig 5.11(b), and any point over a set of ellipses can be selected when $n = 1$ (see Section 5.2.2). This sets the minimum requirement to thwart signal cancellation to three. For the proposed bootstrapping protocol, it is expected that at least three verifiers (e.g., the hub plus two other legitimate devices) will be available, as our work targets a group setting. If not, auxiliary devices can be added for verification purposes. We emphasize that there is no need for an authenticated channel between any auxiliary device and legitimate device.

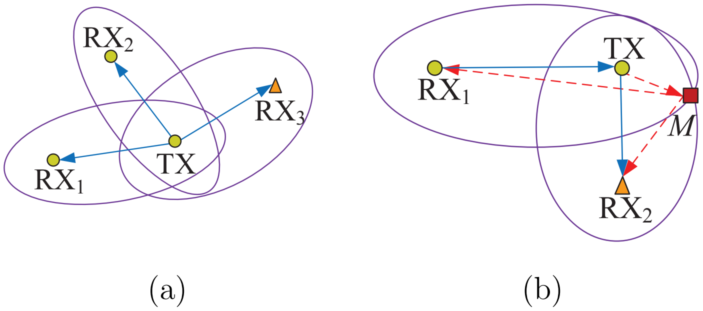
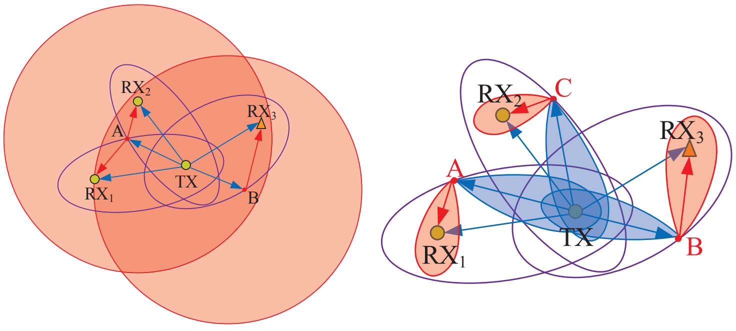


Figure 5.11: (a) TX placed on the shared focus of three ellipses which have RX$_1$, RX$_2$ and RX$_3$ on the other foci respectively. An adversary positioned on one ellipse can cancel the TX signal at the RX positioned at the ellipse's other focal point. No common intersection point exist among three ellipses, and (b) M is placed on the intersection point between two ellipses to simultaneously cancel the signal at RX$_1$ and RX$_2$.

Signal cancellation by a multi-device adversary: A multi-device adversary may be capable of canceling a transmission at more than two RXs. To scale this attack to more RXs, the adversary can deploy additional devices that lie on the intersection of the respective ellipses defined by TX-RX pairs. For instance, Figure 5.12(a) shows the deployment of two devices to perform cancellation at RX$_1$, RX$_2$ and RX$_3$. The device at location A targets at RX$_1$ and RX$_2$, whereas from B to RX$_3$.

However, such a coordinated attack poses significant challenges. First, the transmission of the cancellation signal at location A contaminates the reception of the TX's signal at location B. The latter is necessary to compute the cancellation signal for RX$_3$. Second, the cancellation signal at locations A and B superimpose at RX$_1$ and RX$_2$, thus significantly degrading the cancellation capability. This multi-device attack can be successful only if the interference caused by multiple cancellers is minimal, which is only possible with close placement to the respective RXs when omnidirectional antennas are used. Such a close placement may be apparent to the user.

A higher-cost approach for performing cancellation to multiple RXs without causing unintended interference is to deploy devices with highly directional antennas.



(a) (b)

Figure 5.12: (a) The adversary place two colluding devices one at A with omnidirectional transmission antenna and highly directional receiving antenna and other at B with highly directional antenna, the attack fails due to self-interference, and (b) the adversary places three colluding devices at A, B and C with highly directional antenna.

This scenario is depicted in Figure 5.12(b). Three devices are deployed at locations A, B, and C. Each device is equipped with two directional antennas. One is pointed to the TX to receive the transmitted signal and the other is pointed to the RX to perform cancellation. For a group of n verifiers, $2n$ directional antennas are needed. For a typical device separation of 10-30 ft. with an adversary located at a distance of 60 ft. he is required to achieve 9°-26° beamwidth. Such narrow beamwidths can be created by an antenna array [104] or a parabolic antenna [105]. A 9° beamwidth or a 26° beamwidth antenna array requires approximately 30 antenna elements or 17 antenna elements, respectively.

Our scheme does not provide protection against a multi-device adversary that can perfectly cancel MC ON-OFF sequences with highly-directional non-interfering transmissions from devices located at ideal locations. For all practical purposes, such a potent adversary is in full control of multiple wireless channels and can erase/inject any message at will.

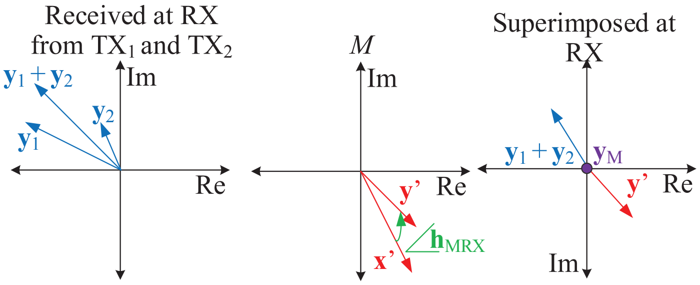


Figure 5.13: Superimposition of signals received from TX$_1$, TX$_2$ and M at RX. M must be able to relay $-y_1 - y_2$ form a single location.

5.4.2 Signal Cancellation from Multiple TXs to One RX

We now consider the inverse scenario where an MC ON-OFF message m is synchronously transmitted by n TXs and is received at a single RX. For this scenario, we examine whether signal cancellation at the RX is possible. A key observation for this case is that although the n TXs convey the same ON-OFF message m, ON slots are realized using different and randomly selected symbols at each TX. Therefore, Let $\mathbf{x}_i = \{x_i(1), x_i(2), \ldots, x_i(k)\}$ denote the transmitted symbols from one TX$_i$ modulating m and $\mathbf{y}_i = \{y_i(1), y_i(2), \ldots, y_i(k)\}$ the received symbols at RX. To cancel the incoming signal at RX, M has to transmit the inverse signal,

$$\mathbf{x}' = -\frac{\sum_{i=1}^{n} \mathbf{h}_{\mathrm{TX}_i\mathrm{RX}} \mathbf{x}_i}{\mathbf{h}_{M\mathrm{RX}}} = -\frac{\sum_{i=1}^{n} \mathbf{y}_i}{\mathbf{h}_{M\mathrm{RX}}}. \tag{5.6}$$

The superposition of $\sum_{i=1}^{n} \mathbf{y}_i$ and $\mathbf{y}'$ for two TXs is shown in Figure 5.13. According to (5.6), the computation of $\mathbf{x}'$ requires the knowledge of the transmitted signals $\mathbf{x}_i$ from all the TXs and of the channels $\mathbf{h}_{\mathrm{TX}_i\mathrm{RX}}$ and $\mathbf{h}_{M\mathrm{RX}}$. However, the adversary does not have knowledge of the randomly transmitted symbols by each TX in advance. Moreover, it receives the superposition of the $\mathbf{x}_i$s, modified by the individual channels. For successful cancellation irrespective of the values of the $\mathbf{x}_i$s, the adversary must be positioned such that it cancels each individual $\mathbf{x}_i$.

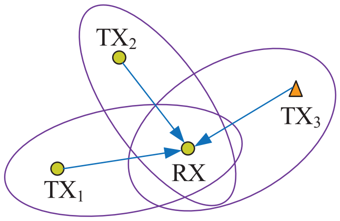


Figure 5.14: RX placed on the shared focus of three ellipses which have TX$_1$, TX$_2$ and TX$_3$ on the other foci respectively. An adversary positioned on one ellipse can cancel the TX signal at the RX positioned at the ellipse's other focal point. No common intersection point exist among three ellipses.

For example, consider the scenario of Figure 5.14, where TX$_1$, TX$_2$, and TX$_3$ transmit $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ respectively and RX receives $\mathbf{y}$ as the superposition of $\mathbf{y}_1$, $\mathbf{y}_2$ and $\mathbf{y}_3$. As this superposition randomly changes with each transmitted symbol, to cancel any ON slot at RX, the adversary must find a location ℓ_M such that it can simultaneously annihilate $\mathbf{y}_1$, $\mathbf{y}_2$, and $\mathbf{y}_3$ by relaying the received signal.

Similarly to the case of one TX and multiple RXs, the adversary must attempt to cancel the symbols from each individual transmission, such that the aggregated symbol is canceled at RX. For each TX, M's location must satisfy the phase difference equation (5.4). The solution to each individual equation is an ellipse with the respective TX and RX located at the focal points of the ellipse. Therefore, ℓ_M must lie in the intersection of three ellipses, as shown in Figure 5.14. These ellipses have RX as a common focal point, with TX$_1$, TX$_2$, and TX$_3$ being the other three focal points. However, Proposition 7 states that no such common intersection point exists. Hence, an adversary cannot find a valid location to perform cancellation from three TXs to one RX. Similarly to the case of one TX and multiple RXs, there are some degenerate TX arrangements that make cancellation from a single location possible. For the case of signal cancellation from multiple TXs to one RX the same complexity arguments as in the previous section. The best approach for the

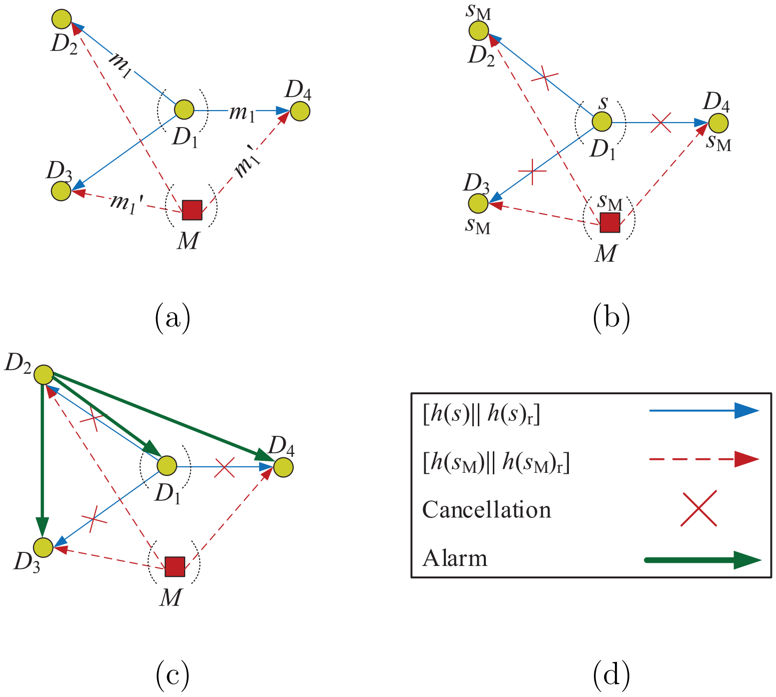


Figure 5.15: (a) M replacing m_1 with m_1' during overshadowing attack, (b) M attempting to perform signal cancellation on D_1's transmission to D_2, D_3 and D_4 during the verification phase of the VERSE primitive (c) D_2 raises the alarm after detecting error during the verification phase of the VERSE primitive, and (d) legends for the figure.

adversary is to cancel the signal of each TX individually using highly directional antennas to avoid unintended interference. The number of devices that need to be deployed grows linearly to the number of legitimate devices.

5.4.3 Security Analysis of the VERSE Primitive

The security of the VERSE primitive is derived from the difficulty in canceling a signal of one TX at multiple verifiers when the number of verifiers is greater than two and canceling the signal from more than two TXs at one verifier. We discuss a basic scenario with three verifiers for each transmission (four devices in total). In this example, M attempts to inject m_1' while D_1 transmits m_1 and pass the verification

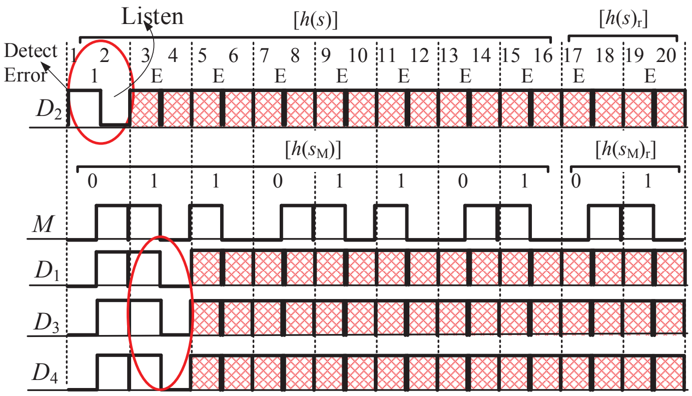


Figure 5.16: M performing signal manipulation attack on D_2's transmission to flip bits where $[h(s) \mid\mid h(s)_r] \neq [h(s_M) \mid\mid h(s_M)_r]$ to pass the verification. D_2 followed by D_1, D_3 and D_4 transmits "Alarm" or error bits by sending all ON slots after detection of energy during its OFF slot.

at the other three devices D_2, D_3, and D_4. The adversary must be capable of injecting m_1' at D_2, D_3, and D_4 simultaneously. This can be achieved by launching an overshadowing attack [54], as shown in Figure 5.15(a). Because m_1 is not ON-OFF modulated and a signal cancellation is not necessary, the adversary can inject a signal with large enough energy that causes demodulation to a desired constellation point. This is plausible for low order constellations (e.g., BPSK, QPSK), where the received constellation point needs to fall within a specific plane or quadrant. Note m_is are not protected with MC ON-OFF keying to improve the time efficiency of the bootstrapping process.

According to the VERSE primitive, D_2, D_3, and D_4 compile $s_M = m_1'||m_2||m_3||m_4$, whereas D_1 compiles $s = m_1||m_2||m_3||m_4$. During the integrity verification phase of VERSE, D_1 transmits $[h(s) \mid\mid h(s)_r]$, while D_2, D_3, and D_4 transmit $[h(s_M) \mid\mid h(s_M)_r]$. To prevent an alarm at D_2, D_3, and D_4, the adversary has to perform signal cancellation on D_1's transmission to replace $[h(s) \mid\mid h(s)_r]$ with $[h(s_M) \mid\mid h(s_M)_r]$ at all the three verifiers. This attack is shown in Figure 5.15(b). However, in Section 5.4.1, we showed that it is infeasible to perform such signal cancellation at more than three verifiers.

Since the adversary is unable to perform signal cancellation on D_1's signal, at least one of D_2, D_3, and D_4, will detect the error when $[h(s) \mid\mid h(s)_r] \neq [h(s_M) \mid\mid h(s_M)_r]$ and raise an alarm. In Figure 5.15(c), we show D_2 raising an alarm during the verification phase. This alarm will be now heard by the rest of the devices because the adversary is not positioned to cancel the signal from D_2 to the remaining three devices. The sequential raising of an alarm by each of the devices is shown in Figure 5.16. We note that even if the adversary is positioned such that it can achieve cancellation to a subset of devices, it cannot cancel the raised alarms as the number of TXs raising alarms increase because it is infeasible to perform signal cancellation from more than two TXs to one RX. There might be other attack vectors where the adversary chooses to overshadow a different combination of messages during the protocol execution phase. For instance, for the scenario of four devises, it could choose to inject m_1' only at D_3 and D_4. In this case, D_1 and D_2 compile s, whereas D_3, and D_4 compile s_M. Hence, to pass the verification the adversary has to perform signal cancellation on the transmissions from D_1 and D_2 to D_3 and D_4 and replace $[h(s) \mid\mid h(s)_r]$ with $[h(s_M) \mid\mid h(s_M)_r]$.

To guarantee the secure operation of VERSE under any possible attack vector we need to have *at least three verifiers for any direction.* This can be achieved by requiring at least four legitimate devices and the hub participate in the group (a total of five devices). Then, irrespective of the set of devices selected by M to perform the overshadowing attack, M will have to perform signal cancellation attack from at least one TXs to at least three RXs, or from at least three TXs to at least one RX. We have shown that neither of these attacks is feasible, due to the impossibility of finding a location to concurrently perform successful cancellation at multiple verifiers.

Even though we have that cancellation attacks to multiple RXs or from multiple TXs are theoretically infeasible, in practice, such attacks could have some limited success probability. This is because the adversary does not have to completely annihilate the incoming signal at a given verifier, but has to reduce it below the detection threshold for an ON slot. This threshold is typically larger than the receiver sensitivity, to account for ambient noise from other devices. Therefore, there could be some location for which M has a cancellation probability p_n for each slot. To guarantee the security of VERSE, we use the length of the hash value used for integrity verification to drive the overall success probability for M to negligible values. This is formalized in the following proposition, where we show that the probability of M successfully modifying any (or multiple) message(s) without being detected by all the legitimate devices is bounded by δ.

Proposition 8. *For a group of size N, the VERSE is δ–secure against message modifications with*

$$\delta \;\leq\; \left(p_H + (1-p_H)p_n\right)^{\ell}, \tag{5.7}$$

where δ is the probability that M can replace any m_i sent by D_i with m_i' at any subset of remaining devices without being detected at every $D_{i'} \in \mathcal{D}$ (where $\mathcal{D}$ is the set of all legitimate devices), p_H is the probability for a bit of $h(s)$ to equal a bit of $h(s_M)$, and p_n is the probability of successfully flipping one bit in $[\cdot]$ during transmissions from n TXs to one RX or from one TX to n RXs where $n = \lceil N/2 \rceil$, and ℓ is the length of the hash function $h(\cdot) \mid\mid h(\cdot)_r$. We show that δ is a negligible function of ℓ.

Proof. Let's consider an adversary that targets to modify one message m_i sent by D_i[1]. In the simplest case, the adversary replaces m_i with m_i' at all other legitimate devices $\mathcal{D} \setminus D_i = \mathcal{D}_{-i}\{D_{i'} | i' \neq i\}$, where $\mathcal{D}$ denotes the set of all legitimate devices in the group. During the VERSE verification phase, all the $D_{i'}$ compiles $s_M = m_1 || \dots || m_i' || \dots || m_n$, whereas D_i compiles $s = m_1 || \dots || m_i || \dots || m_n$. Then to pass the transcript verification M has to replace $[h(s) \mid\mid h(s)_r]$ with $[h(s_M) \mid\mid h(s_m)_r]$ at all the $D_{i'}$ on transmission from D_i, so that none of the verifiers raise an alarm. If any

[1] Modifying multiple messages is more difficult, in which case the success probability is upper bounded by that of modifying a single message.

one other verifier $D_{i'}$ raises an alarm, then all the others will detect the MitM attack and raise an alarm, since a single M can only be set to cancel the transmissions from one TX (D_i) to other RXs at one time, but not from $D_{i'}$ to those RXs. Hence, the adversary has to perform signal cancellation on transmission of one TX to multiple (all other) RXs in this case.

In general, M might choose to replace m_i with m_i', at a subset of other legitimate devices, $\mathcal{D}_M = \{D_{i'}|, i' \in 1, 2, ...N, i' \neq i\} \subset \mathcal{D}_{-i}$, such that during the VERSE verification phase D_i and all the $D_{i''} \in \mathcal{D}_{-i} \setminus \mathcal{D}_M$ compile the same communication transcript as s, whereas every $D_{i'} \in \mathcal{D}_M$ compiles s_M. Then to pass the transcript verification M has to replace (cancel and inject) $[h(s) \mid\mid h(s)_r]$ with $[h(s_M) \mid\mid h(s_m)_r]$ at all the $D_{i'} \in \mathcal{D}_M$ on transmissions from D_i and every $D_{i''} \in \mathcal{D}_{-i} \setminus \mathcal{D}_M$, and vice versa, to replace the ON-OFF signals from $D_{i'} \in \mathcal{D}_M$ to all devices in $D_{i''} \in \mathcal{D}_{-i} \setminus \mathcal{D}_M$ and D_i, such that none of the verifiers raise the alarm. Hence, the adversary has to perform signal cancellation on transmissions of multiple TXs to multiple RXs simultaneously.

In any of the above cases, the success of the adversary is upper-bounded by the capability to replace $[h(s) \mid\mid h(s)_r]$ with $[h(s_M) \mid\mid h(s_m)_r]$ on transmission from one TX to multiple RXs, or from multiple TXs to one RX. Next, we compute the probability of replacing $[h(s) \mid\mid h(s)_r]$ with $[h(s_M) \mid\mid h(s_m)_r]$. First, we compute the probability that the k^{th} bit is received as $h(s_M) \mid\mid h(s_M)_r^k$ at all $D_{i'} \in \mathcal{D}_M$ (say, from D_i). This occurs if one of the following two conditions is met: either the k^{th} bit is the same in $h(s) \mid\mid h(s)_r$ and $h(s_M) \mid\mid h(s_M)_r$ or M is able to perform cancellation and injection of k^{th} at all $D_{i'} \in \mathcal{D}_M$:

$$\begin{aligned}
\Pr[k^{th} = h(s_M)^k] &= \Pr[h(s)^k = h(s_M)^k] + \\
&\qquad \Pr[h(s)^k \neq h(s_M)^k] \Pr[\text{Cancel}] \\
&= p_H + (1 - p_H)p_n, \qquad (5.8)
\end{aligned}$$

where p_H is the probability for a bit of $h(s) \mid\mid h(s)_r$ to equal a bit of $h(s_M) \mid\mid h(s_M)_r$, and p_n is the probability upper bound of successfully flipping one bit in $[\cdot]$ during transmissions from multiple TXs to one RX or from one TX to multiple RXs (it is reasonable to assume the same p_n applies to both scenarios).

For a strictly universal hash function, the hashes for two different inputs differ at each bit with probability $1/2$. The probability δ of accepting the modified message of M at A is computed by taking into account the total number of bits (ℓ) generated by the hash function $h(\cdot) \mid\mid h(\cdot)_r$. The adversary's modified message is accepted by all the $D_{i\prime}$ if M has replaced m_i with m_i' and $[h(s_M) \mid\mid h(s_M)_r]$ is received at all $D_{i'}$ instead of $[h(s) \mid\mid h(s)_r]$. We argue that successful cancellation of every ON-slot occurs independently, as each ON slot symbol transmitted by each device is randomly generated (i.i.d). This is because, if the attacker is located at a fixed location, the resulted aggregated signal relayed by the attacker will be randomly distributed (and independent across symbols), so the probability of each aggregated received symbol's power being less than a threshold is also independent from each other. Thus, δ is the product of the probability of successfully manipulating each bit:

$$\begin{aligned}
\delta &\leq \Pi_{k=1}^{\ell} \Pr[k^{th} = h(s_M)^k] \\
&\leq \Pi_{k=1}^{\ell}(p_H + (1-p_H)p_n) \\
&\leq (p_H + (1-p_H)p_n)^{\ell}. \qquad (5.9)
\end{aligned}$$

where p_H is the probability for a bit of $h(s) \mid\mid h(s)_r$ to equal a bit of $h(s_M) \mid\mid h(s_M)_r$, and p_n is the probability of a successfully flipping one bit in $[\cdot]$ during transmissions from multiple TXs to one RX or from one TX to multiple RXs, and ℓ is the length of the hash function $h(\cdot) \mid\mid h(\cdot)_r$. It is easy to show that δ is a negligible function of ℓ, since $p_H + (1-p_H)p_n < 1$ (as long as $p_n < 1$ in general, for any number of verifiers). Since for each possible sub-case (of adversary choosing to modify one message from any device to any subset of remaining devices), we have the same success probability bound δ, we can conclude that the adversary's overall success probability is also upper bounded by δ, meaning with probability at least $1-\delta$, all

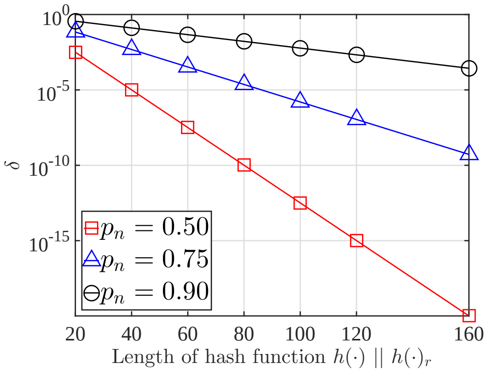


Figure 5.17: The probability that M can replace m_i with m_i' without being detected at $D_{i'}$ $\forall$ $i' \neq i$.

the devices in the group will detect the MitM attack. □

Note that the above proposition is general and applies to any group size $N > 1$. However, for different values of N, we have different concrete guarantees since p_n depends on the minimum number of devices n (number of transmitters for many-to-one, or receivers for one-to-many) that the adversary needs to launch a cancellation attack against, among all possible cases of group partitioning. For example, when $N = 2$, the minimum number of cancellation targets is 1; in general, for $N \geq 5$, $n = \lceil N/2 \rceil$. In addition, according to our experiments in Section VI, we show that for $n = 1, 2$, p_n can be as large as 0.9. However, p_n dramatically drops to a very small value when $n = 3$. Thus, the security guarantee of the VERSE primitive is stronger with an increasing n and also the group size N.

Figure 5.17 shows δ as a function of the hash length ℓ for various values of p_n when $p_H = 0.5$ (i.e., the bits of the $h(s)$ and $h(s_M)$ are random). As expected, a higher p_n yields higher δ values for the adversary. For instance, when $p_n = 0.9$ we have $\delta = 0.00027$ for $\ell = 160$. But when the cancellation probability is significantly low, for instance when $n = 3$, $p_n = 8.7 \times 10^{-5}$, we have $\delta = 6.9 \times 10^{-49}$ for $\ell = 160$. We note that this is an *online* attack that has to be performed while the pairing session is ongoing security. Similar standards are used for other existing pairing protocols [27]. Moreover, a $p_n = 0.9$ is difficult to achieve in the presence of multiple D_i's. δ is a negligible function of ℓ, the adversary's success probability can always be driven to any desired value by choosing a long enough ℓ.

5.4.4 Security of the Bootstrapping Protocol using VERSE

We now analyze the security of the bootstrapping protocol shown in Figure 3.3 against MitM attacks, which can be reduced to the security of VERSE (Corollary 1). Basically, we need to show that the adversary can neither join the group as an additional device and pair with any existing legitimate device nor can the adversary carry out an MitM attack against any legitimate device(s) to pair itself with the hub A or any D_i.

Corollary 1. *The bootstrapping protocol protected by the VERSE primitive is $\delta-$secure against active attacks with*

$$\delta \;\leq\; \left(p_H + (1 - p_H)p_n\right)^{\ell}. \tag{5.10}$$

Here, δ is the probability that M can replace any DH public number m_i (sent by any device or A) with m_i' at any subset of remaining devices, without being detected at every device $D_{i'} \in \mathcal{D}$ (including the hub). Notations are defined in the same way as in Proposition 8.

Proof. The only differences between our bootstrapping protocol and the VERSE primitive are: (a) the addition of an initialization phase, where the devices are synchronized and the group count is pre-loaded to A, and (b) the messages being exchanged are the DH public numbers. The message content does not affect the security because of hash function's collision-resistance property. We analyze the security of the bootstrapping protocol in two parts. First, we address the case of a malicious device attempting to pair with the legitimate hub. We then analyze the case where a rogue hub attempts to pair with a legitimate device. Note that, an adversary targeting the synchronization phases of the protocol will fail to pair with either the legitimate hub or devices, as we will show in Proposition 9 later. In the following we assume that the adversary does not attempt a desynchronization attack.

Malicious device pairing with the legitimate hub: Any malicious device that simply participates in the protocol will appear as an additional device beyond the $N - 1$ legitimate devices indicated by the user. The extra device count leads to the abortion of the protocol according to Step 3. The legitimate hub raises an alarm by broadcasting all ON slots during the MC ON-OFF transmission of the protocol transcript digest. As we showed in Proposition 7, this broadcast cannot be canceled and eventually propagates to all legitimate devices.

An alternative approach for the adversary would be to hijack the pairing session of a legitimate device so that the total number of participating devices is not violated. The integrity verification phase prevents this hijacking because the transmission of the protocol transcript digest is protected by the VERSE primitive. According to Proposition 8, as long as any subset of devices computes different transcripts, all devices will detect the attack with probability no less than $1 - \delta$.

Rogue hub pairing with a legitimate device: The adversary can attempt to pair with a legitimate device by posing as the hub and hijacking the pairing session with the legitimate hub. To carry out this attack against a device D_i, the adversary has to perform a signal overshadowing attack and replace the legitimate DH primitive m_A with m_M at D_i. Moreover, the adversary has to replace the protocol transcript digest $[h(s) \mid\mid h(s)_r]$ transmitted by the remaining legitimate devices and A to D_i, with $[h(s') \mid\mid h(s')_r]$. Proposition 8, states that as long as any subset of devices computes different transcripts, all devices will detect the attack with probability no less than $1 - \delta$. Hence, the adversary will fail to pose as a legitimate hub. □

Moreover, in Proposition 9, we show that an adversary targeting the initialization phase to either desynchronize the legitimate devices or make them synchronize with a rogue hub leads to a protocol failure. Therefore, we do not need to introduce a secure synchronization mechanism.

Proposition 9. *The bootstrapping protocol protected by VERSE fails under a desynchronization attack during the initialization phase.*

Proof. The attack on the synchronization between the legitimate entities during the simultaneous MC ON-OFF transmission can be mitigated by initiating VERSE simultaneously at all the legitimate entities. We first discuss the initialization step, followed by the security analysis of it. Finally, we discuss the security analysis on the attack on synchronization.

According to Step 1 of the pairing protocol, the protocol is initiated by the user by powering ON all the legitimate devices and setting the hub to pairing mode. This step is followed by the transmission of the DH primitives. To inform each device when all other devices are powered ON and ready to pair, we have added the coordination process.

Initially, the user sets the hub to pairing mode by pressing a button on the hub device. When in this mode, the hub broadcasts a random MC ON-OFF sequence while waiting for other devices to be turned ON. This mode lasts for a pre-specified time period τ sufficient for pairing all other devices, or until the users press the pairing button again. This phase terminates by transmitting a known delimiter (ON-ON-OFF-OFF-ON-ON). When legitimate devices are powered ON, they listen to the ON-OFF sequence broadcasted by the hub and wait for the known delimiter to synchronously initiate Step 2. Note that the known delimiter further allows the devices to time synchronize with the clock of the hub.

To combat possible active attacks on initialization and/or time synchronization, each device remains in pairing mode for a period τ' which is slightly longer than τ, even if it has already paired with the hub.

We now demonstrate that an adversary targeting the initialization and/or synchronization of the protocol will fail to pair with the legitimate hub or a legitimate device. Consider the device activation sequence shown in Figure 5.18. Because the

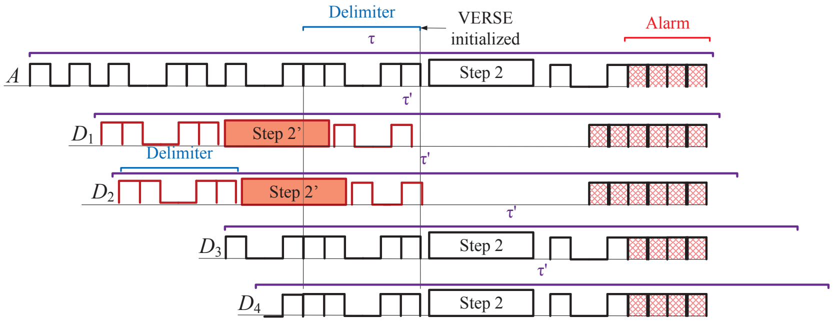


Figure 5.18: Attack on the initialization step of VERSE.

delimiter used to denote the end of the initialization phase is public, an adversary can attempt to pair with a legitimate device by performing a signal cancellation and injection attack. In this attack, the adversary cancels the ON-OFF sequence of the hub and injects a delimiter sequence to cause the initiation of the pairing process sooner than the time intended by the legitimate hub. According to Proposition 7, the adversary is able to cancel the ON-OFF sequence at most at two devices, say D_1 and D_2. These two devices may complete the pairing process with the malicious hub before other legitimate devices are activated or execute the protocol with the legitimate hub. However, they remain in pairing mode for a period $\tau' > \tau$.

When devices D_3 and D_4 execute the VERSE protocol with the legitimate hub, the adversary has to replace the expected messages from D_1 and D_2 with his own messages to satisfy the group count. This can be done by a simple message injection. However, during the confirmation stage, all devices synchronously transmit the ON-OFF sequence of the protocol transcript digest. In our example, at least A, D_3, and D_4 will transmit that sequence. As a result, D_1 and D_2 will overhear a second integrity verification phase (Step 3) within their pairing period τ'. Based on Proposition 7, the adversary cannot perform cancellation from three transmitters to one receiver to prevent the overhearing of the legitimate confirmation phase at D_1 and D_2. The two latter devices will raise an alarm by transmitting all ON slots during the integrity verification phase and the protocol will terminate in FAILURE.

This delimiter is sent by the hub before the synchronous transmission of the protocol digest is initiated (Step 3). We clarify that we have not assumed a secure synchronization protocol between the hub and the legitimate devices. We have simply stated that under a benign setting, the devices are capable of achieving synchronization with a bounded error ϵ. This error has been assumed to be fairly large in our experimentations relative to typical clock drifts of wireless devices and topology scenarios considered in this work (we set ϵ between 1μs to 30μs). Such a value demonstrates that VERSE operates correctly even in worst-case time misalignment scenarios. If the adversary attacks the second SYNC message to misalign the legitimate transmitters, the ON-OFF sequence transmitted during the integrity verification phase will be misaligned leading to the sounding of the alarm by transmitting all ON slots. Therefore, the adversary cannot successfully join the group, by causing time misalignment between legitimate devices.

Now we will present the security analysis on the attack of synchronization between legitimate entities. Two attack scenarios can weaken the security of the proposed group pairing protocol: (a) a malicious device pairs with the legitimate hub, or (b) a legitimate devices pairs with a rogue hub.

Malicious device pairing with the legitimate hub: The device synchronization is initiated by the hub, by sending the delimiter message in Step 3, when the VERSE primitive is used to secure the transmission of $[h(s) \mid\mid h(s)_r]$. To pair with hub A, the malicious device must follow the timing set by the end of the delimiter sent from A. Any message received by A at a different timeline will be aborted. The adversary can attempt to cancel the delimiter message sent by A at a target device D_i, so as to prevent D_i from broadcasting the protocol digest with other devices. The goal is to reduce the number of devices where cancellation should take place when $[h(s) \mid\mid h(s)_r]$ is transmitted using ON-OFF mode by the remaining devices. However, device D_i will overhear the MC ON-OFF sequence transmitted by the remaining of devices, without having received the delimiter. This sequence from many simultaneous transmitters to one receiver cannot be canceled by the adversary.

Device D_i will raise an alarm by transmitting continuous ON slots, leading to the protocol failure. So attacking the synchronization protocol can only lead to a DoS and does not provide the adversary with an additional capability to compromise the protocol.

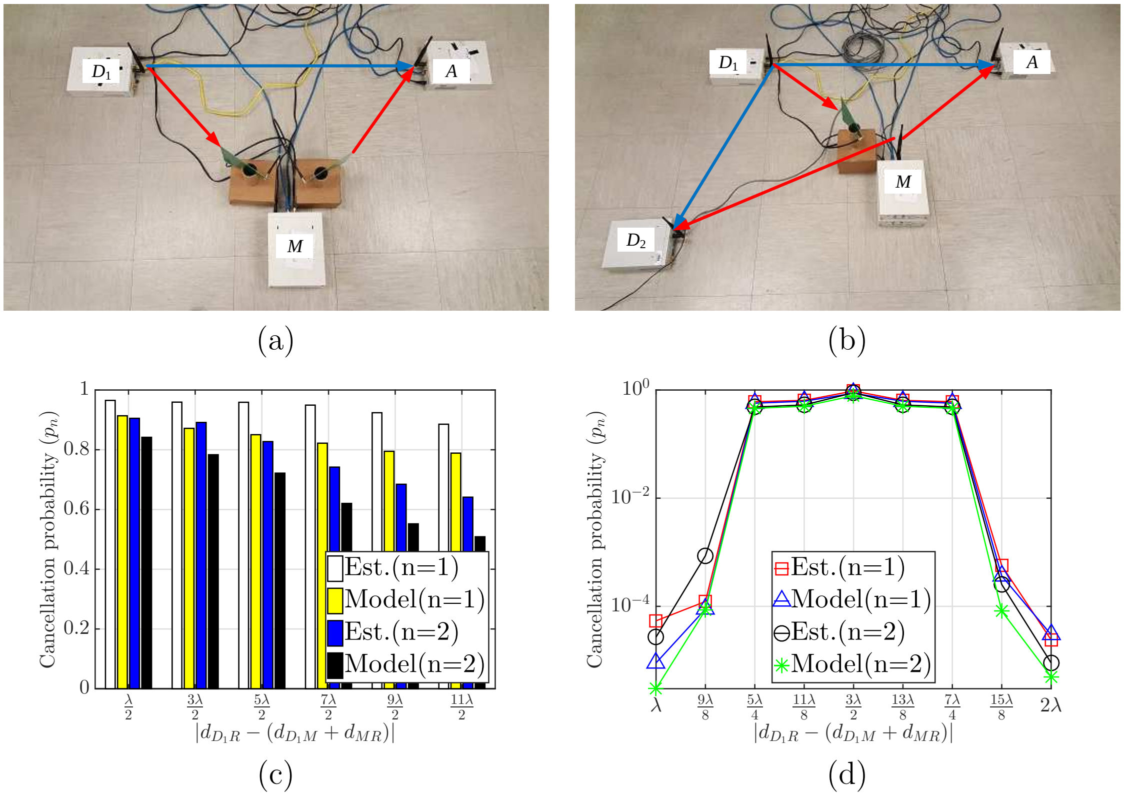
Malicious device posing as a legitimate hub: The adversary can also attempt to synchronize the legitimate device to his own delimiter message rather than the legitimate hub. If the desynchronized device transmits when the MC ON-OFF sequence of the protocol transcript is transmitted by legitimate devices, the legitimate devices and the hub will detect energy during the OFF slots and abort the protocol.

This proves that the VERSE is protected against any attack on the synchronization between the legitimate entities. □

5.5 Evaluation

In this section, we experimentally evaluate the effectiveness of signal cancellation under different number of verifiers. We also discuss practical implementation details.

Experimental Setup: We performed all the experiments using NI-USRP 2921 devices. Each device and the hub was realized by one USRP device. The adversary was implemented using two USRP devices one for listening and one for relaying. The listening adversarial device was equipped with a directional antenna (LP0965 Log Periodic PCB Antenna, 850MHz to 6.5GHz) aimed at the TX, whereas the adversarial transmitting device was equipped with either a directional antenna aiming at one RX, or an omnidirectional antenna targeting multiple RXs. All devices were synchronized (with the clock of the same computer) and transmitted at 2.4GHz with 22MHz bandwidth. The slot duration was fixed to 1ms. An ON slot was realized with the transmission of 250 random symbols with 4μs duration, whereas an OFF slot was realized with silence. Experiments were performed at night to minimize Wi-Fi interference although Wi-Fi beacon signals were present during the experiments. The threshold for determining an ON slot was set to -50dBm, which is significantly



(a) (b) (c) (d)

Figure 5.19: (a) Experimental setup for signal cancellation from one TX to one RX, (b) experimental setup for signal cancellation from one TX to two RXs, (c) cancellation probability as a function of the distance difference between the direct and the relay paths when M is placed at an ellipse satisfying eq. (5.4), and (d) cancellation probability as a function of the distance difference between the direct and the relay paths, when M is perturbed from the location with path difference equal to $3\lambda/2$.

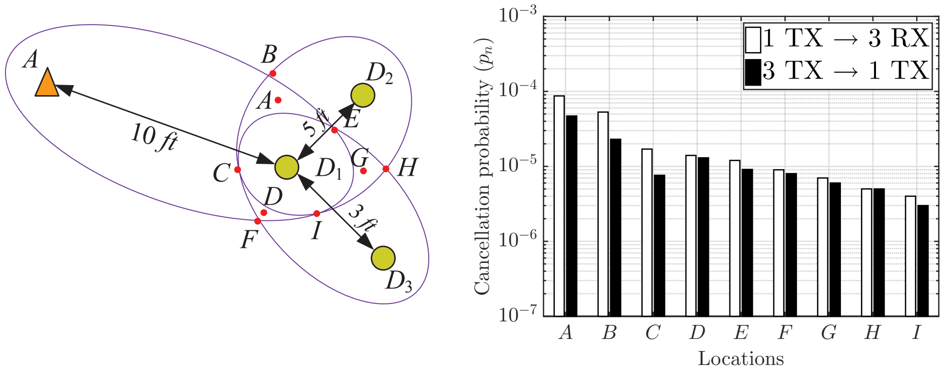
higher than the receiver sensitivity (typically at or less than -70dBm). This higher value was selected to minimize false positives due to ambient wireless activities at the 2.4GHz band. Each experiment was repeated 10^6 times.

5.5.1 Effectiveness of the Signal Cancellation

Signal cancellation when $n = 1, 2$. In the first set of experiments, we evaluated the probability p_n (used in Proposition 8 and Corollary 1) of successful signal cancellation via a relay attack for $n = 1$ and $n = 2$. For $n = 1$, we used the experimental setup shown in Figure 5.19(a). A device D_1 sent 10^6 MC ON-OFF modulated bits to a hub A in the presence of M who performed a relay cancellation. The two USRPs implementing M were stacked on top of each other at a location on one ellipse that satisfied (5.3) and (5.4). For $n = 2$, we used the experimental setup shown in Figure 5.19(b). The adversary was placed at the intersection of the two ellipses that satisfied (5.3) and (5.4). The transmitting antenna of M was replaced with an omnidirectional one to allow the simultaneous cancellation at two locations.

The receiver at M played three roles: (a) estimate the respective channels, (b) quickly detect ON slots using energy detection, and (c) determine the symbols being transmitted from D_1 during ON slots in an online fashion as M is not aware of the pseudo-random symbols transmitted by D_1. The estimated channel was used to craft the amplitude of the symbol relayed by M's transmitter to cancel D_1's signal at the receivers (the phase was matched based on M's location). The transmitting signal at M was crafted using two approaches. In the first approach, M estimated the $\mathbf{h}_{D_1M}$ and $\mathbf{h}_{MA}$ channels based on the transmissions of D_1 and A, respectively. The $\mathbf{h}_{D_1A}$ channel was modeled after a Rician model with a K factor equal to two, which represents an indoor environment with a strong LoS component. In the second approach, no channel estimation took place at M. All channels were modeled after a free-space path loss model with an attenuation exponent $\alpha = 2$.

Figure 5.19(c) shows the cancellation probability (p) as a function of the difference between the direct and relay paths. The adversary was placed at the different ellipses dictated by eq. (5.4), and for w =1, 2, 3, 4, 5, and 6. We observe that when the adversary is close and therefore, has a dominant LoS channel to D_1 and H, the cancellation probability is quite high (94.56% and 91.17% for estimated channel and modeled channel attenuation, respectively for $n = 1$ and 90.57% and 84.42% for estimated channel and modeled channel attenuation, respectively for $n = 2$). Even at several wavelengths away, signal cancellation remains possible with non-negligible probability. The cancellation performance is worse for $n = 2$ because M



(a) (b)

Figure 5.20: (a) Experimental topology for the evaluation of security primitive of VERSE, and (b) cancellation probability for the experimental setup of (a).

has to perform simultaneous cancellation at both A and D_2 and more channels need to be estimated. Moreover, the channel estimation yields a stronger cancellation capability compared to channel modeling for both $n = 1$ and $n = 2$.

Sensitivity to location placement: In the next set of experiments, we studied the sensitivity of the cancellation attack to M's location. The adversary was placed at a set of ellipses with a path difference between λ to 2λ and incremented by a step of $\lambda/8$. Figure 5.19(d) shows the cancellation probability as a function of the difference between the direct and relay path. As expected, the cancellation probability is maximized when the path difference equals ($3\lambda/2$), which satisfies eq. (5.4). The cancellation probability drops significantly when M's location deviates more than $\lambda/2$ from the optimal location for both $n = 1$ and $n = 2$. From this experiment, we verify that signal cancellation attacks are sensitive to the adversary's location due to the short wavelength of the carrier frequency. A location perturbation of just a few centimeters is sufficient to reduce the effectiveness of the attack, as M's signal no longer arrives at the targeted RXs with the opposite phase.

Signal cancellation when $n = 3$: We also evaluated the signal cancellation capability for the one TX/three RX scenario and the three TX/one RX scenario.

These two cases serve as the basis for the security of VERSE. We used the topology shown in Figure 5.20(a). In the first scenario, D_1 broadcasted MC ON-OFF signals that were simultaneously received by three RXs. According to Proposition 7, there is no single location that allows M perform signal cancellation to all three RXs. Therefore, we selected a set of locations that could likely succeed in canceling some of the received signals. Specifically, the adversary is placed in all locations marked by dots. Locations (A, B, C, E, F, H) correspond to the intersection of two ellipses whereas locations (D, G, I) are the centroids of the areas created by the three closest intersection points. In the second scenario, A, D_2, and D_3 synchronously transmitted an MC ON-OFF signal that was received by D_1.

Figure 5.20(b) shows the cancellation probability for the two different scenarios and for each location. We observe that for any scenario, the cancellation probability is below 10^{-4}. Moreover, the cancellation probability was non-zero in all cases due to the relatively high threshold value (-50dBm) that was used to detect ON slots. Although the adversary's signal was not the exact inverse to annihilate legitimate transmissions, on certain occasions, there was sufficient alignment to drop the received power below -50dBm for the respective RX(s). It should be noted here that this experiment is not the proof of the adversary's inability in performing cancellation when $n > 2$, but the proof is derived from Proposition 7.

Alarm raising probability: We further evaluated the security of VERSE in terms of raising an alarm. We replicated the experimental setup of Figure 5.20 and implemented the verification phase where every device transmits the hash of the protocol transcript using MC ON-OFF modulation. We considered an adversary that successfully replaced m_1 of D_1 with m'_1 leading to the compilation of s_M at D_2, D_3, A and the compilation of s at D_1. To account for a varying number of bits that must be canceled by M, we varied the Hamming distance between $h(s_M)$ and $h(s)$ from 0.1 of the hash length (160 bits) to 0.8 of the hash length. This is done by randomly generating two 160-bit strings with the desired Hamming distances. An alarm was raised by any device that detected a transmitted sequence different than

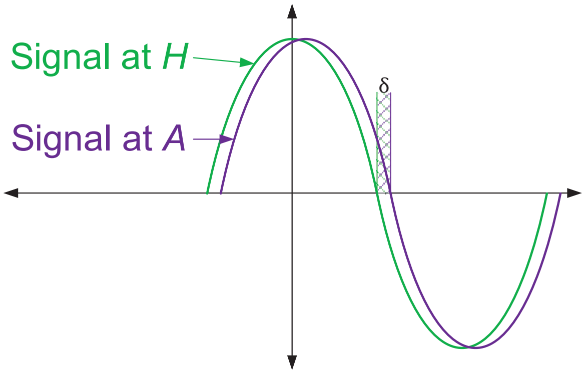


Figure 5.21: An example of superimposed received signals from D_1, D_2 and D_3 which are misaligned by an offset of ϵ.

the one it was transmitting. In all scenarios tested and for all adversary locations, *all verifiers* detected the message manipulation and raised an alarm. The attack was detected with probability one for all 10^6 hash transmissions.

5.5.2 Practical Considerations

We now analyze the time synchronization requirement, interference effect for the VERSE protocol and its timing overhead.

Synchronization: During the verification phase of VERSE, multiple devices must simultaneously transmit an ON-OFF sequence. Possible misalignment between the clocks of each device may lead to false alarms. To address the possible time misalignment, the hub broadcasts a delimiter just before the start of the verification phase, to synchronize The clock of each device. Despite this synchronization, there is still possible time misalignment between the devices due to clock drift and the different path delays caused by multipath or NLoS channels to each receiver. There have been extensive studies on synchronization of independent wireless nodes [90], but practically it is impossible to reach perfect synchronization.

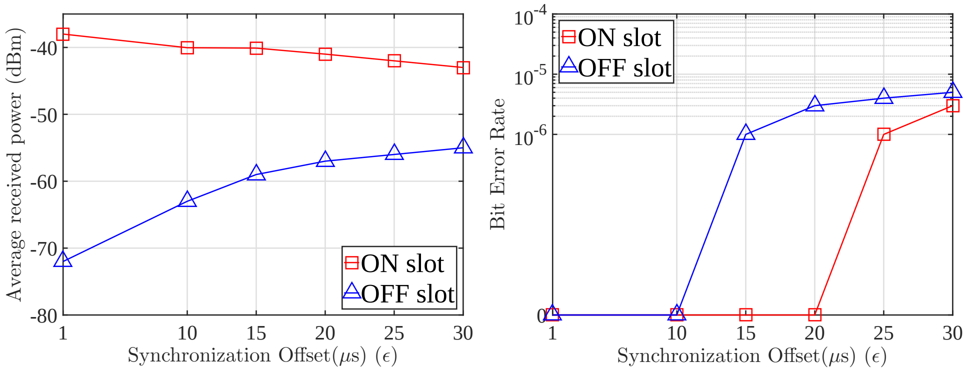
Figure 5.21 shows an example, where D_1, D_2, and D_3 transmit simultaneously, with the transmissions being misaligned by a time offset ϵ. Misalignment causes some energy from ON slots "bleed" into OFF slots and some silent period of the OFF slot "bleed" into ON slots. However, the offset ϵ is much smaller (a few μsec) than the slot duration for the ON-OFF sequence which is set to 1ms. The state $s(j)$ of the j^{th} slot is decided according to the following rule:

$$s(j) = \begin{cases} \text{OFF}, & \text{if } p(j) \leq \gamma_D, \\ \text{ON}, & \text{if } p(j) > \gamma_D. \end{cases} \tag{5.11}$$

where γ_D is the detection threshold (set to -50dBm in our experiments), and $p(j)$ is the average received power over the j^{th} slot. To resolve the time misalignment problem, a solution similar to [114] can be adopted. Rather than averaging the power of all the samples in slot j, an RX eliminates the samples corresponding to an interval $\epsilon_{\max}$ from the beginning and the end of each j^{th} slot, where the slot boundaries are computed according to the RX's own clock. This strategy leaves a time interval of $T - 2\epsilon_{\max}$ for estimating the received power, where T is the slot duration, and, $\epsilon_{\max}$ is the maximum time offset between any of the devices.

Experimental evaluation of synchronization: We set up three USRP devices to transmit ON-OFF messages simultaneously, while a fourth USRP was acting as the intender RX. We placed the TXs that simultaneously transmitted the random MC ON-OFF sequence at different locations in the laboratory with both LoS and NLoS channels to the RX. TX$_1$ was placed behind a bookshelf inside the room, TX$_2$ was placed outside the room to ensure an NLoS channel, whereas TX$_3$ was placed at a LoS to the RX. The transmit power for an ON slot was set to 20dBm with a symbol duration of 1ms. An artificial clock misalignment from $\epsilon = 1\mu$s to $\epsilon = 30\mu$s was induced between D_1, D_2, and D_3 to emulate the maximum time offset error. The experiment lasted for the transmission of 10^6 sequences of 40 bits each.

The first experiment was performed to select the detection threshold γ_D. Figure 5.22(a) shows that average received power during an ON slot varied from -42dBm



(a) (b)

Figure 5.22: (a) Average received power of superimposed signal from D_1, D_2, and D_3 on ON and OFF slots as a function of synchronization offset (ϵ), and (b) bit error rate as a function of synchronization offset (ϵ) in μs.

to -38dBm. The received power during an OFF slot varied from -72dBm to -55dBm indicating the presence of some ambient noise. The detection threshold was set to $\gamma_D = -50dBm$.

In the second experiment, we used the same setup with experiment one and evaluated the slot detection error rate as a function of the synchronization offset. To cope with the time misalignment, the RX excluded the first 30μs from the beginning and end of each slot. The results for the ON slot error rate and the OFF slot error rate are shown in Figure 5.22(b). We observe that ON slots are always correctly detected for any time offset. For the OFF slots, a very small number (seven slots out of 10^6) were wrongly estimated. This indicates that excluding the samples at the beginning and end of each slot effectively addresses the synchronization problem.

Interference Effect: To make VERSE robust to interference from co-existing wireless systems, we set the detection threshold for ON slots significantly higher than the typical receiver sensitivity. In the experiments, we selected the detection threshold for ON slots to be -50dBm, which is orders of magnitude higher than the average noise level (typically at -120dBm). The security of VERSE could be impacted because the adversary no longer has to cancel a transmission to the noise floor, but achieving cancellation below the detection threshold is sufficient. To account for this tradeoff, the system security, as expressed by Proposition 8 and Corollary 1, incorporate the probability p_n of successfully flipping a bit during cancellation. This probability parametrizes the success of the adversary in performing cancellation due to considering a higher than the noise floor detection threshold.

Timing Analysis: The timing overhead of VERSE includes the following components (a) the initialization step, (b) exchanging the public DH parameters, and (c) transmitting in MC ON-OFF mode the digest of the protocol transcript. The initialization step can be maximum of τ for powering of all the group devices by the user so they can be set to pairing mode, which can be set to 120s [115]. From the remaining two components, the verification phase dominates the protocol's timing performance, since the ON-OFF mode is significantly slower than nominal transmission speeds. However, the ON-OFF keying time is constant to the group size. For a hash with length ℓ, a total of $2(\ell + r)$ slots of duration T are necessary to complete the verification phase. Assuming typical values of $\ell = 256$, $r = 256$ (in the worst case) and $T = 1ms$ [31], the verification phase requires 1.024s to complete which is acceptable for all practical uses and it is independent of the number of participating devices.

5.6 Chapter Summary

We addressed the problem of securely bootstrapping a group of devices to a hub when none of the devices share any prior security associations. We propose VERSE, a new PHY-layer group message integrity verification primitive resistant to MitM attacks over the wireless channel. We exploit the existence of multiple devices that act as verifiers of the protocol transcript for integrity protection. When three or more devices perform an integrity check, it is infeasible for the adversary to simultaneously manipulate the wireless signal at all devices, based on geometrical constraints. We presented a DH-based device bootstrapping protocol that utilized VERSE, which only requires in-band communications with minimal human effort during initialization. We formally prove the security of both VERSE and the bootstrapping protocol against active attacks. With a real-world USRP testbed, we experimentally validated our theoretical results by showing that an increasing number of devices significantly weakens the adversary's ability to successfully manipulate wireless signals. This is in contrast to prior state-of-the-art where the attacker's success probability increases with the number of devices.

CHAPTER 6

VERIFYING ADS-B NAVIGATION INFORMATION THROUGH DOPPLER SPREAD MEASUREMENTS

6.1 Introduction

6.1.1 Motivation

The International Air Transport Association has forecasted that over 7.8 billion passengers will use air transport annually by 2036 [42]. To cope with the anticipated increase in air traffic, the relevant governing bodies around the world have agreed to a novel air traffic control technology that shifts traffic surveillance from the uncooperative and independent radar system to a cooperative and dependent digital one. At the heart of this new technology lies the Automatic Dependent Surveillance-Broadcast (ADS-B) standard [43]. In ADS-B, aircraft independently determine their navigation information (location, airspeed, heading, etc.) using onboard satellite equipment (GPS) [13]. To facilitate air traffic management, this navigation information is broadcasted to nearby aircraft and ground air traffic control (ATC) centers.

ADS-B is expected to significantly reduce the cost of traffic control, as radar systems are expensive to deploy and maintain. Moreover, it will improve aviation safety by delivering fine-grained navigation information in a timely fashion. Due to its profound advantages, many aviation carriers have already introduced ADS-B equipment into their air fleet [116–119]. Despite its critical function, ADS-B does not integrate strong security mechanisms. The aircraft's location is verified by ground stations using a multilateration technique. In this technique, three or more ground stations compute the time difference of arrival from an ADS-B broadcast to validate the claimed aircraft's position [13]. However, an aircraft has no way of verifying the ADS-B broadcast of another aircraft.

Researchers have highlighted and even implemented numerous attacks that can be launched with COTS equipment and rudimentary knowledge [11–13, 120–122]. Costin et al. [13] have experimentally demonstrated the feasibility of replay/injection attacks on the ADS-B using USRPs and COTS equipment. Sampigethaya et al. [11] have enumerated various threats in ADS-B such as eavesdropping, radio-frequency jamming, aircraft impersonation, active manipulation of data, etc. In fact, none of the fundamental security properties, namely source authentication, data integrity, data confidentiality and resistance to jamming can be guaranteed under the present standard. Consequently, ADS-B transmissions can be eavesdropped, spoofed, replayed, modified, deleted, and jammed [12, 13, 120, 122].

The ADS-B security vulnerabilities can be abstracted to classical cryptography problems for which solutions are readily available [122]. These solutions require the introduction of cryptographic primitives. However, implementing cryptographic solutions at a global scale requires coordination between multiple governing agencies, administrators and operators. Key management operations including key establishment, key refresh, key revocation, certificate management, etc. introduce a substantial layer of complexity and cost to the ADS-B standard [123]. Moreover, any recommended changes to the current ADS-B standards, require extensive retrofitting and upgrade efforts for the already deployed ADS-B equipment. Such changes involve universal software updates for introducing security modules or even hardware updated if the deployed solutions require secure hardware.

To cope with these challenges, we examine non-cryptographic solutions for verifying the navigation information broadcasted in ADS-B. We make the following contributions.

6.1.2 Main Contributions and Chapter Organization

We address the problem of verifying the integrity of ADS-B navigation information *without modifying the ADS-B standard.* We develop a PHY-layer based method for verifying the aircraft position and velocity advertised in unencrypted ADS-B frames, by exploiting the Doppler spread phenomenon. We show that a malicious ground station cannot spoof a "ghost" aircraft by transmitting ADS-B frames containing rogue navigation vectors. Defeating our verification method is equivalent to forging signatures in unbalanced oil and vinegar signature schemes [124].

Chapter Organization: The remainder of this chapter is organized as follows. In section 6.2 gives a brief overview of ADS-B architecture. In Section 6.3, we state the problem. The method for verifying the integrity of aircraft velocity and position is presented in Section 6.4. In Section 6.5, we evaluate our method via simulations. Related work is described in Section 6.6 and in Section 6.6.1, we conclude.

6.2 ADS-B Architecture Overview

The ADS-B standard regulates the exchange of broadcast messages between aircraft and ATC ground stations. An entity can operate as a transmitter, referred to as ADS-B OUT, or as a receiver, referred to as ADS-B IN (see Figure 6.1). At the PHY layer, periodic navigation broadcasts are transmitted using either the 978 MHz Universal Access Transceiver (UAT) data link or the 1090 MHz Extended Squitter (1090ES) data link [125]. The suggested range for ADS-B transmissions reaches the 90 nautical miles for aircraft-to-aircraft communication and 150 nautical miles for aircraft-to-ATC communication [125]. It is suggested that ADS-B frames are transmitted every 0.5 sec.

ADS-B frames are modulated with pulse-position modulation (PPM), with a pulse length of 1μs. Therefore, ADS-B achieves a data rate of 1 Mbps. ADS-B frames consist of an 8.0μs long preamble used for frame synchronization and a 56/112 bit payload. The various fields of the payload are shown in Figure 6.2. DF

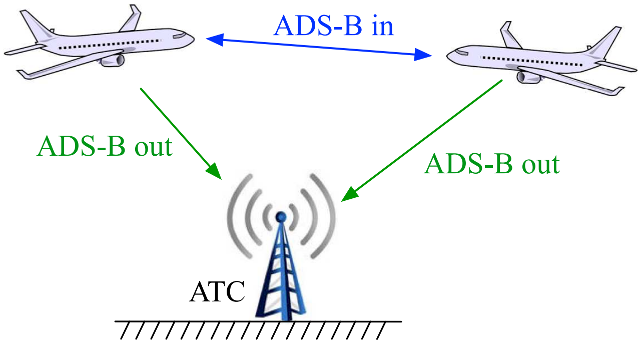


Figure 6.1: The ADS-B architecture.

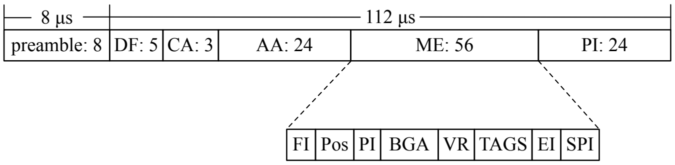


Figure 6.2: The ADS-B frame format.

refers to the downlink format used to encode broadcast messages. CA indicates if capability 17 is set for 1090ES. The AA field contains the 24-bit globally unique ICAO aircraft address. The aircraft navigation information is contained within the 56 bit-long ME field. Finally, the last 24 bits contain a CRC for detecting and correcting errors.

The ME field consists of the following subfields: (a) flight identification (flight number call sign)(FI), (c) position (latitude/longitude)(POS), (d) position integrity/accuracy (GPS horizontal protection limit)(PI), (e) barometric and geometric altitudes (BGA), (f) vertical rate (rate of climb/descent) (VR), (g) track angle and ground speed (velocity), (TAGS) (h) emergency indication when the emergency code is selected (EI), and (i) special position identification when IDENT is selected (SPI).

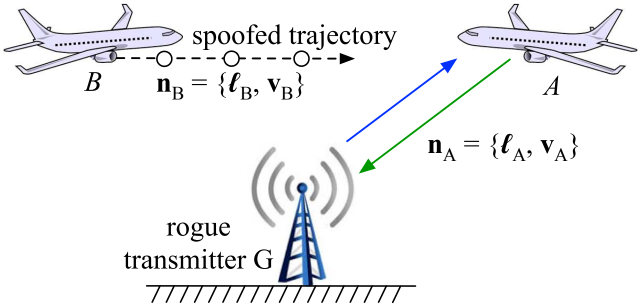

spoofed trajectory
B
$\mathbf{n}_B = \{\boldsymbol{\ell}_B, \mathbf{v}_B\}$
A
$\mathbf{n}_A = \{\boldsymbol{\ell}_A, \mathbf{v}_A\}$
rogue
transmitter G

Figure 6.3: Spoofing aircraft B at A by transmitting ADS-B messages from G.

6.3 Problem Statement and Assumptions

We consider the scenario depicted in Figure 6.3. An aircraft A with navigation information $\mathbf{n}_A=\{\boldsymbol{\ell}_A, \mathbf{v}_A\}$ is within the range of a rogue ground station G. For simplicity, the navigation information of A contained in the ME field of an ADS-B frame is abstracted to a position vector $\boldsymbol{\ell}_A$ with Cartesian coordinates $\{x_A, y_A, z_A\}$ and a velocity vector $\mathbf{v}_A$. The rogue ground station attempts to spoof the existence of a ghost aircraft B with navigation information $\mathbf{n}_B^{cl}$, by transmitting a crafted ADS-B compliant signal from a static location[1] $\boldsymbol{\ell}_G$. Spoofing of B is targeted specifically at A, whose navigation information is known at G. We address the problem of enabling A, who acts as the *verifier*, to reject the $\mathbf{n}_B^{cl}$ transmitted by G, who acts as the *prover*. We only consider solutions that do not require modifications to the ADS-B standard. As a result, cryptographic mechanisms that verify source authenticity and message integrity cannot be employed. Such mechanisms would require the re-standardization of the ADS-B protocol, costly redeployment efforts, and the establishment of a worldwide key management system.

6.4 Velocity and Position Verification

In this section, we propose a verification method for validating the velocity and position claims included in ADS-B frames. The central idea of our method is to

[1] We do not consider possible spoofing attacks from airborne adversaries.

exploit the Doppler spread phenomenon for measuring the relative radial velocity between the verifier and the prover. We show that it is difficult to manipulate the maximum Doppler spread measurements performed by the verifier. Using the relative radial velocity, a verifier aircraft A can check both the velocity and position claims of a prover aircraft B, which are connected through well-defined kinematic equations. Figure 6.4 shows the relationship of the position and relative radial velocity between two aircrafts A and B at k distinct locations. Specifically, let the magnitude of the relative radial velocity $|\mathbf{v}_{B|A}|$ between A and B be:

$$|\mathbf{v}_{B|A}| = |\mathbf{v}_A - \mathbf{v}_B| \cos\theta, \tag{6.1}$$

where $|\mathbf{x}|$ is the magnitude of vector $\mathbf{x}$, and

$$\cos\theta = \frac{\boldsymbol{\ell}_A \cdot \boldsymbol{\ell}_B}{|\boldsymbol{\ell}_A||\boldsymbol{\ell}_B|}, \tag{6.2}$$

is the angle of the line connecting A and B. The maximum Doppler spread ω_D measured at A is proportional to $|\mathbf{v}_{B|A}|$.

$$\omega_D = \frac{2\pi|\mathbf{v}_{B|A}|f_c}{c}, \tag{6.3}$$

where f_c is the carrier frequency and c is the signal propagation speed. Hence, given $f_c, c, \boldsymbol{\ell}_A, \boldsymbol{\ell}_B$, and $\mathbf{v}_A$, estimating $\mathbf{v}_B$ is equivalent to estimating ω_D. Doppler spread estimation has been extensively used in wireless communications for improving functions at the PHY layer (adaptive coding, modulation, antenna diversity, power control, handoff [126–128]).

6.4.1 Maximum Doppler Spread Estimation

Several methods have been proposed for estimating the maximum Doppler spread [129–131]. For our purposes, we have selected the method proposed by Tepedelenlioğlu et al. [131] because (a) it is shown to be more accurate for high-velocity

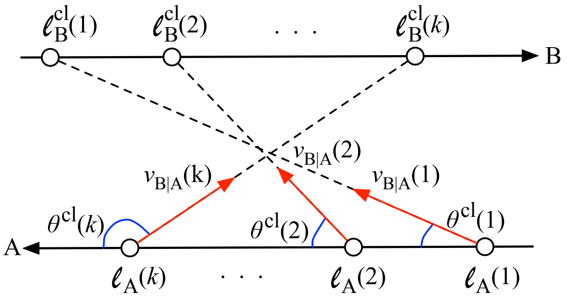


Figure 6.4: Position and relative radial velocity at k distinct locations.

vehicles, (b) it is robust to additive white noise, and (c) it relies on channel measurements that are difficult to manipulate and predict. We briefly describe the estimator in [131], which uses the I/Q components of the channel response $h(t)$ to measure ω_D through

$$\omega_D = \sqrt{\frac{-2r_h''(0)}{r_h(0)}}, \tag{6.4}$$

where $r_h(\tau) = E[h(t) * h(t+\tau)]$ is the autocorrelation function for the channel $h(t)$ and $r_h''(0)$ is the second derivative of r_h at zero. These values are obtained by the following steps.

Step 1: Compute $M+1$ channel correlation estimates

$$\{\hat{r}_h(iT)\}_{i=0}^{M}, \tag{6.5}$$

by sample averaging the channel $h(t)$ with a sampling period T. That is,

$$\begin{aligned} \hat{r}_h(iT) \quad &= \quad \frac{\sum_j h(jT)h((j+i)T)}{\alpha N_s - i}, \\ & \quad\quad j = 0, \ldots, (\alpha N_s - i) \end{aligned} \tag{6.6}$$

The channel samples $h(jT)$ are computed by sampling N_s symbols of a known signal (e.g., frame preamble). The value $\alpha = \frac{T_s}{T}$ denotes the number of samples collected per sampled symbol, when the symbol duration is T_s. Note that the values of M

and T are selected such that $MT << 1$.

Step 2: Compute matrix

$$A = (L^T L)^{-1} L^T R_H, \tag{6.7}$$

where

$$A = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}, \quad L = \begin{bmatrix} 0^0 & 0^1 & 0^2 \\ 1^0 & 1^1 & 1^2 \\ \vdots & \vdots & \vdots \\ M^0 & M^1 & M^2 \end{bmatrix},$$

$$R_H = \begin{bmatrix} \hat{r}_h(0) \\ \hat{r}_h(T) \\ \vdots \\ \hat{r}_h(MT) \end{bmatrix}.$$

Step 3: Estimate $r_h(0)$ and $r''_h(0)$ from:

$$\left\{ r_h^{(n)}(0) = n! a_n / T^n \right\}, \quad n = 0, 2, \tag{6.8}$$

where a_0 and a_2 are obtained from Step 2.

Step 4: Substitute $r''_h(0)$ and $r_h(0)$ in (6.4) to estimate ω_D.

6.4.2 Verification Process

Using the estimated ω_D, the verifier A computes the magnitude of its relative radial velocity to the prover B. The relative radial velocity is used to verify the claims of the prover. The verification steps executed by A are as follows.

Step 1: Verifier A receives k ADS-B frames from prover B and records the claimed positions and velocities included in the payload:

$$\mathcal{L}_B^{cl} = \{\boldsymbol{\ell}_B^{cl}(1), \boldsymbol{\ell}_B^{cl}(2), \ldots, \boldsymbol{\ell}_B^{cl}(k)\}. \tag{6.9}$$

$$\mathcal{V}_B^{cl} = \{\mathbf{v}_B^{cl}(1), \mathbf{v}_B^{cl}(2), \ldots, \mathbf{v}_B^{cl}(k)\}. \tag{6.10}$$

Step 2: For all claimed positions, the verifier A calculates the claimed headings Θ^{cl}:

$$\Theta^{cl} = \{\theta^{cl}(1), \theta^{cl}(2), \ldots, \theta^{cl}(k)\}, \tag{6.11}$$

$$\theta^{cl}(i) = \cos^{-1} \frac{\boldsymbol{\ell}_A(i) \cdot \boldsymbol{\ell}_B^{cl}(i)}{|\boldsymbol{\ell}_A(i)||\boldsymbol{\ell}_B^{cl}(i)|}. \tag{6.12}$$

Step 3: The verifier A estimates $\mathcal{V}_{B|A}^{est}$ for all received k frames using the maximum Doppler spread estimation method.

$$\mathcal{V}_{B|A}^{est} = \{|\mathbf{v}_{B|A}^{est}(1)|, |\mathbf{v}_{B|A}^{est}(2)|, \ldots, |\mathbf{v}_{B|A}^{est}(k)|\}. \tag{6.13}$$

Step 4: The verifier A estimates the velocity of B for each of the k received ADS-B frames using $\mathcal{V}_{B|A}^{est}$ and its own velocity.

$$\mathcal{V}_B^{est} = \{|\mathbf{v}_B^{est}(1)|, |\mathbf{v}_B^{est}(2)|, \ldots, |\mathbf{v}_B^{est}(k)|\}. \tag{6.14}$$

Step 5: The verifier A computes the normalized root mean square error for the velocity estimator:

$$RMSE_v = \sqrt{\frac{\sum_i \left(\frac{|\mathbf{v}_B^{cl}(i)| - |\mathbf{v}_B^{est}(i)|}{|\mathbf{v}_B^{est}(i)|}\right)^2}{k}}, \ i = 1, \ldots, k. \tag{6.15}$$

In (6.15), the difference in magnitude between the claimed and estimated velocities is normalized to the magnitude of the velocity estimated via the maximum Doppler spread method.

Step 6: The verifier A calculates the estimated and claimed distance covered in interframe time t_P, using kinematic equations:

$$d^{est}(i) = \frac{|\mathbf{v}_B^{est}(i)| + |\mathbf{v}_B^{est}(i-1)|}{2} * t_P, \tag{6.16}$$

$$d^{cl}(i) = \boldsymbol{\ell}_B^{cl}(i) - \boldsymbol{\ell}_B^{cl}(i-1), \quad i = 2, \ldots, k. \tag{6.17}$$

Step 7: The verifier A computes the normalized root mean square error for the distance

$$RMSE_\ell = \sqrt{\frac{\sum_i \left(\frac{d^{cl}(i) - d^{est}(i)}{d^{est}(i)}\right)^2}{k}}. \tag{6.18}$$

In (6.18), the difference in magnitude between the distance covered in t_p is normalized to the magnitude of the distance estimated from the relative radial velocity.

Step 8: If the $RMSE_v \leq \gamma_v$ and $RMSE_\ell \leq \gamma_\ell$, then accept $\mathcal{V}_B^{cl}$ and $\mathcal{L}_B^{cl}$. Else, reject them.

We emphasize that in Step 6, we employed kinematic equations modeling straight-line trajectories for aircraft flying at a constant velocity. This model is valid when aircraft fly at cruising speed, given the small duration of the verification process (a few seconds). However, this model may not be accurate during takeoff and landing. For the latter, a more complex flight trajectory model can be employed. In this work, we focus on demonstrating the potential of exploiting the PHY-layer attributes on the verification process, rather than exhausting all possible aviation situations. The number of ADS-B frames k necessary for robust verification and the threshold values γ_v, γ_ℓ are system parameters that are empirically tuned depending on the aviation scenario. We study the impact of both parameters in Section 6.5.

6.4.3 Security Analysis

In this section, we examine if a stationary rogue station G can spoof the trajectory of a ghost aircraft B while passing the verification process presented in the previous section. We examine two possible spoofing methods. In the first method, G selects a desired trajectory represented by $\mathcal{L}_B^{cl}$ and crafts ADS-B frames that prove $\mathcal{L}_B^{cl}$ to the verifier A. In the second method, G estimates the maximum Doppler spread measured by the verifier A at k positions and attempts to find a valid trajectory $\mathcal{L}_B^{cl}$ that satisfies the estimated relative radial velocities.

Spoofing a desired trajectory $\mathcal{L}_B^{cl}$: Let G transmit k ADS-B frames, claiming position and velocity sets $\mathcal{L}_B^{cl}$ and $\mathcal{V}_B^{cl}$, respectively. First, note that sets $\mathcal{L}_B^{cl}$ and $\mathcal{V}_B^{cl}$ are not independent, but are bound by the kinematic equations (6.16) and (6.17). By fixing $\mathcal{L}_B^{cl}$, the claimed positions translate to a set of relative radial headings Θ_B^{cl} according to (6.2). Computation of these headings requires the knowledge of the trajectory of A. The latter can be predicted based on the navigation information broadcasted by A, assuming a straight line trajectory with constant velocity during the expected broadcast of the k ADS-B frames by G. From Θ_B^{cl}, the rogue station G computes the magnitude of the relative radial velocities $\mathcal{V}_{B|A}^{cl}$ that needs to be estimated by A to pass the verification process. The $\mathcal{V}_{B|A}^{cl}$ translate to a set of maximum Doppler spread measurements using (6.3).

The problem of spoofing the desired trajectory $\mathcal{L}_B^{cl}$ reduces to the problem of spoofing a set of maximum Doppler spread values at A. However, the maximum Doppler spread depends on the h_{GA} channel, which is not under the control of G. The only way that G can influence the estimation of h_{GA} at A is by modifying the preamble $x(t)$ of the ADS-B frames. We now show that spoofing ω_D^{sp} by altering $x(t)$ to $x'(t)$ is hard.

Let us consider the transmission of a single ADS-B frame and a desired ω_D^{sp} to be measured at A. The rogue station G can estimate $h_{GA}(t)$ using the ADS-B frames broadcasted by A ($h_{GA}(t)$ and $h_{AG}(t)$ can be considered equivalent due to the channel reciprocity principle[2]). G can then alter the amplitude and phase of the preamble $x(t)$ to $x'(t)$, so that A estimates a desired channel $g_{GA}(t)$ instead of

[2] The channel reciprocity principle primarily holds for low-Doppler spread channels. However, several methods exist to compensate for RF impairments in other cases [132].

$h_{GA}(t)$.

$$x'(t) = \frac{h_{GA}(t)}{g_{GA}(t)} x(t). \tag{6.19}$$

The problem of backtracking the maximum Doppler spread estimation method becomes equivalent to finding the channel samples $g(jT)$ at A that yield the desired ω_D^{sp}. This can be attempted via the following steps:

Step 1: Fix $r_h(0)$ to any value between 0 and 1. Using the known $r_h(0)$, ω_D^{sp}, f_c and c, calculate $r_h''(0)$ from equation (6.4).

Step 2: Calculate the values of a_0 and a_2 from $r_h(0)$ and $r_h''(0)$, respectively, using equation (6.8). Fix a_1 to any value between 0 and 1. This yields matrix

$$A = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}.$$

Step 3: Using A and L, calculate $R_H = A(L^T L)(L)^{-1}$.

Step 4: R_H yields the $M' = M + 1$ correlation estimates $\hat{r_h}(iT)$ that need to be computed by A when sampling $g_{GA}(t)$. Each $\hat{r_h}(iT)$ is calculated by averaging over αN_s channel samples (see eq. (6.6)). This forms the following system of M' equations and αN_s unknowns, which are the samples of the desired $g_{GA}(t)$ at A.

$$(S) \begin{cases} \sum_j g(jT)g((jT) - \alpha N_s \hat{r_h}(0) = 0 \\ \vdots \\ \sum_j g(jT)g((j+M)T) - \alpha N_s \hat{r_h}(MT) = 0 \end{cases} \tag{6.20}$$

The equations in S form a multivariate underdefined quadratic equation system. The general problem of solving such systems is NP-hard [124], with the best-known algorithms performing almost equivalently to exhaustive search, even for small values of M' [124]. The problem difficulty has motivated their use in public cryptosystems.

Specifically, the so called "Unbalanced Oil and Vinegar" (UOV) scheme is thought to be secure if $3M' \leq \alpha N_s \leq \frac{M'(M'+2)}{2}$ [124]. For a system with M' equations and αN_s unknowns, the M' variables are said to be the "oil" unknowns and the $\alpha N_S - M'$ variables are said to be the "vinegar" unknowns. In our setup, the number of preamble symbols N_s and the symbol duration T_s are fixed by the ADS-B standard. Therefore, to satisfy the conditions of a difficult-to-solve UOV system, we control the sampling period T for each preamble symbol and the number of correlation values M' used to estimate the channel in 6.6. These values are fixed such that $3M' \leq \frac{T_s N_s}{T}$ and $\frac{T_s}{T} N_s \leq \frac{M'(M'+2)}{2}$, so that the UOV condition [124] is satisfied.

To spoof the desired Doppler shift ω_D^{sp}, the rogue ground station has to find a solution $\Psi^* = \{g(T_s), g(2T_s), \ldots, g(\alpha N_s T_s)\}$ for S. However, S has many solutions Ψ due to its underdefined nature, with only Ψ^* leading to the computation of ω_D^{sp}. Finding Ψ^* can only be done via exhaustive search, using methods such as the Levenberg–Marquardt algorithm [133]. For a large number of variables, the size of the search space is prohibitive for a timely solution. We emphasize that this spoofing method requires knowledge of the h_{GA} channel for crafting the preamble at G for all k frames. The channel h_{GA} has been assumed to be known based on the reciprocity principle (using the ADS-B transmissions of A). However, the channel coherence time is particularly short (less than 1 ms) due to the high aircraft velocity. Hence, even if G estimates h_{AG} from A's transmissions, the h_{GA} channel is expected to quickly decorrelate from the observed state.

Shifting the central frequency: We now examine if G can spoof a desired trajectory $\mathcal{L}_B^{cl}$ by shifting the central frequency used to transmit the k ADS-B frames. The idea behind this attack is to exploit equation (6.3) used for converting the maximum Doppler spread to the relative radial velocity. Similar to the attack of the previous section, G selects a desired $\mathcal{L}_B^{cl}$ for the ghost aircraft B. By fixing $\mathcal{L}_B^{cl}$, the claimed positions translate to a set of relative radial headings Θ_B^{cl} according to (6.2). From Θ_B^{cl}, the rogue ground station G computes the magnitude of the relative radial velocities $\mathcal{V}^{cl}_{B|A}$ that need to be estimated by A to validate $\mathcal{L}^{cl}_{B}$. The problem of spoofing $\mathcal{L}^{cl}_{B}$ becomes equivalent to finding a set of central frequencies

$$\begin{aligned} \mathcal{F}_c^{sp} &= \{f_c^{sp}(1), f_c^{sp}(2), \ldots, f_c^{sp}(k)\}, \\ f_c^{sp}(i) &= \tfrac{|v^{cl}_{B|A}|(i)}{|v_{B|A}|(i)} f_c, \quad i = 1, \ldots, k. \end{aligned} \tag{6.21}$$

However, equation (6.21) is linear with f_c. To change a true relative radial velocity $|v_{B|A}|$ by $p\%$, the rogue ground station has to shift f_c by $p\%$. Because $f_c = 1090$ MHz, even a small shift in f_c will cause an uncorrectable frequency offset (FO) at A. The ADS-B standard specification requires that receivers can tolerate a FO up to 312.5 KHz [125]. This FO value translates to a possible change in the true relative radial velocity of up to 0.03%. Any larger shifts in the center frequency will render the ADS-B frame undecodable.

Alternate trajectory for true maximum Doppler spread: An alternate strategy for G, is to spoof a trajectory that is compliant with the true maximum Doppler spread measured at A over k ADS-B frames. This strategy is possible because the validation of the prover's trajectory is performed via the magnitude of the relative radial velocity. Therefore, there can exist more than one trajectories that yield the true ω_D's. In this spoofing attack, G first computes the set of relative radial velocities $\mathcal{V}^{est}_{B|A}$ that will be estimated by A, given A's trajectory and G's fixed position. From $\mathcal{V}^{est}_{B|A}$, the rogue ground station G attempts to find a trajectory $\mathcal{L}^{cl}_{B}$ that yields $\mathcal{V}^{est}_{B|A}$ and satisfies the kinematic equations (6.1), (6.16), and (6.17). Specifically, it formulates the following overdefined quadratic equation system.

$$(P) \begin{cases} \mathbf{v}^{est}_{B|A}(i) = |\mathbf{v}_A(i) - \mathbf{v}^{cl}_B(i)| \frac{\boldsymbol{\ell}_A(i) \cdot \boldsymbol{\ell}^{cl}_B(i)}{|\boldsymbol{\ell}_A(i)||\boldsymbol{\ell}^{cl}_B(i)|} \\ \boldsymbol{\ell}^{cl}_B(i) - \boldsymbol{\ell}^{cl}_B(i-1) = \frac{|\mathbf{v}^{cl}_B(i)| + |\mathbf{v}^{cl}_B(i-1)|}{2} * t_P \\ \mathbf{v}^{cl}_B(1) = \mathbf{v}^{cl}_B(2) = \ldots = \mathbf{v}^{cl}_B(k) \end{cases}$$

The system P has $k+1$ unknowns (the k locations in the trajectory $\mathcal{L}^{cl}_{B}$ plus

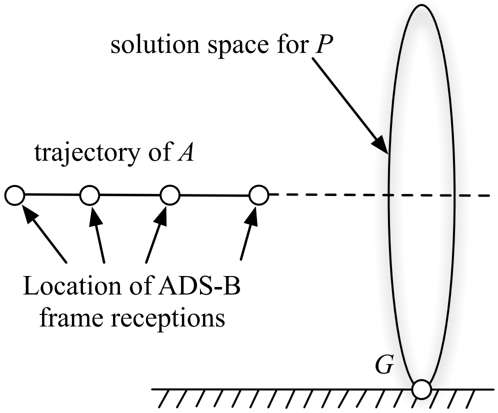


Figure 6.5: The possible spoofed locations of G that satisfy P.

the constant aircraft velocity $\mathbf{v}_B^{cl}$) and $2k - 1$ equations. Finding one solution (but not necessary for all solutions) to a system of multivariate polynomial equations is known to be NP-hard [134]. In general, systems with random equations of this type are not expected to have any solution, and for systems for which one solution is known to exist, other interference solutions are not expected to exist. In our context, at least one solution exists, which yields the true location for G. That is, the trajectory $\mathcal{L}_B^{cl}$ for the spoofed aircraft B degenerates to the static location of G. By symmetry, it is easy to show that any point lying on a circle passing through G and centered at the intersection of A's trajectory with the perpendicular plane, satisfies P (see Figure 6.5). This is because the headings used for the computation of the relative radial velocity based on transmissions from G do not change if G lies at any point of the circle. However, trajectories which degenerate to single points are of little use to the adversary.

6.5 Evaluation

In this section, we evaluate thresholds γ_v and γ_ℓ used in the position and velocity verification. We also demonstrate that truthful location/velocity claims pass the verification process, while spoofed ones are rejected.

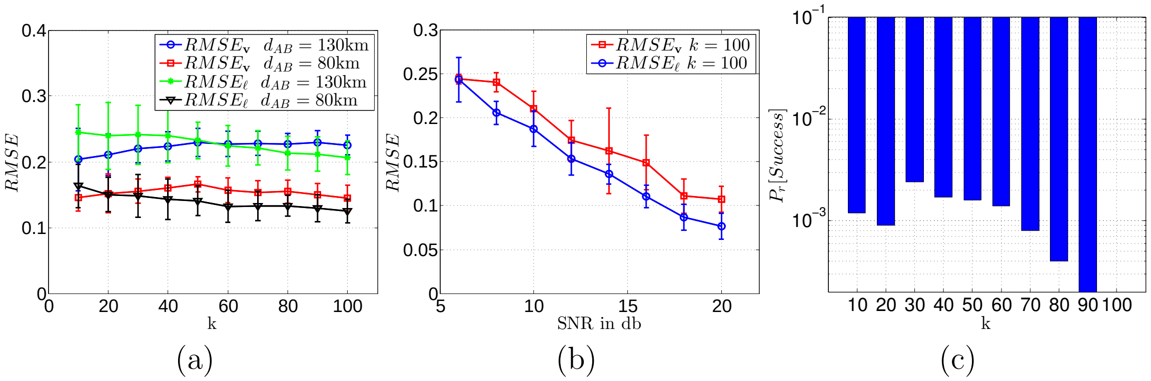


Figure 6.6: (a) $RMSE_v$ as a function of k, (b) $RMSE_\ell$ as a function of k, (c) probability of a successful emulation of a trajectory spoofing attack.

Simulation Setup: We performed our simulations in MATLAB R2014a. Unless otherwise noted, the prover and the verifier were assumed to fly at a constant cruising speed of 900 km/h^{-1} and in opposite directions, while maintaining a constant altitude (as shown in Figure 6.4). The symbol duration was set to $T_s = 10^{-6}$ sec based on the 1 Mbps transmission rate of ADS-B. We simulated a Rician channel $h(t)$ between the prover aircraft and the verifier aircraft. We set the K-a factor of the Rician model to 50, which is appropriate for line-of-sight communications at high altitude and varied the maximum Doppler shift according to the velocities of the aircraft. We used Jake's model to simulate the Doppler spectrum. The channel $h(t)$ was estimated by the verifier using the 8-symbol preamble of ADS-B frames ($N_s = 8$). Finally, we set the sampling period to $T = 5 * 10^{-8}$ sec (20 samples per symbol). For system (S) in (6.20), the selected parameters yield an underdefined system of 16 equations with 16 "oil" variables and 134 "vinegar" variables, which satisfies the required conditions for the security (S) [124].

Scenario 1: First, we considered a benign scenario in which aircraft B proves its true trajectory to aircraft A. We measured $RMSE_v$ and $RMSE_\ell$ as a function of the number of ADS-B frames k used for the verification in (6.15) and (6.18). Two initial distances were considered between the two aircraft; $d_{AB} = 130$ km and $d_{AB} = 80$ km. Figure 6.6(a) shows $RMSE_v$ and $RMSE_\ell$, averaged over 100 repeated experiment executions. Confidence intervals of 95% are also shown. We observe that average $RMSE_v$ and $RMSE_\ell$ values remain relatively constant with k. However, the variance is reduced as k increases, leading to a more robust estimation of the aircraft location and velocity. We further measured $RMSE_v$ and $RMSE_\ell$ as a function of the SNR, when $k = 100$. Figure 6.6(b) shows a decreasing RMSE as the SNR improves. The RMSE plots allow us to select the thresholds γ_v and γ_ℓ used to verify the validity of a location/velocity claim at different SNR regimes, corresponding to verifications occurring at different distances between the prover and the verifier.

Scenario 2: In the second scenario, we considered a rogue ground station G spoofing a ghost aircraft B at aircraft A. To spoof B, we followed the steps of the security analysis presented in Section 6.4.3. We selected a straight line trajectory originating at $d_{AB} = 130$ km away from A, for an aircraft moving in the opposite direction of A at 900 km/h. Based on Figure 6.6(a), we set $\gamma_v = 0.25$ and $\gamma_\ell = 0.3$. We used the trajectories of A and B, we computed the headings and relative radial velocities that need to be spoofed by G. We assumed full knowledge of the h_{GA} channel at G and formed the underdefined equation system (S) in (6.20). To solve (S), we used the in-built MATLAB solver *fsolve*, which employs the Levenberg–Marquardt curve-fitting algorithm [133] to perform an exhaustive search on the solution space. We used the set of targeted relative radial velocities as an input seed into the algorithm. We repeated this process 10,000 times and counted the number of times that G was capable of finding a solution to (S) that would meet both the γ_v and γ_ℓ thresholds. Figure 6.6(c) shows the ratio of the successful spoofing attempts (when both $RMSE_v$ and $RMSE_\ell$ are less than the corresponding thresholds) to the total number of attempts. We denote this ratio as $P_r[Success]$ and plot it as a function of the number of ADS-B frames used in the verification process. Our results show that G can spoof a ghost aircraft with low probability. Moreover, this probability decreases with the number of ADS-B frames used in the verification.

6.6 Related Work

Prior work on the ADS-B security has primarily focused on highlighting vulnerabilities to well-known attacks in wireless communications. Sampigethaya et al. have analyzed the security and privacy of ADS-B in the context of an “e-enabled” aircraft [11]. They defined an adversary model for the aviation domain and enumerated various RF communications related threats. These threats include eavesdropping, radio-frequency jamming, aircraft impersonation, active manipulation of data, and others. They have also proposed a list of system requirements for securing the ADS-B operation.

Strohmeier et al. surveyed ADS-B attacks that have been reported in recent literature [122]. Specifically, they discussed eavesdropping, jamming, message injection, message modification, and message deletion. Moreover, they presented state-of-the-art theoretical and practical efforts to counter the ADS-B threats. McCallie et al. also performed a survey on the vulnerabilities of ADS-B and related these vulnerabilities to air transportation operation and management risks [16]. They classified attacks to a taxonomy based on their nature to facilitate the application of possible solutions.

Costin and Francillon experimentally demonstrated the insecurity of ADS-B using solely the USRP platform and COTS radio transceivers [13]. By implementing a practical, low-cost and moderately sophisticated attacker, they demonstrated ADS-B message replay/injection attacks with relative ease. They also suggested solutions relying on the integration of lightweight cryptographic mechanisms.

While the threats on ADS-B are well-documented, few solutions exist that mitigate such threats. Sampigethaya and Poovendran proposed a group navigation method for verifying the message integrity of ADS-B IN messages. They presented a framework in which aircraft are divided into groups according to average velocity, spatial dependency, and temporal restrictions derived from their trajectories. Each group is coordinated by a leader, who verifies the position of other aircraft by measuring time-difference-of-arrival of ADS-B messages. They further proposed a security simulation tool concept to visualize and asses the impact of ADS-B vulnerabilities.

Several researchers have proposed the integration of cryptographic mechanisms into the ADS-B standard [13,119,135]. Using well-known cryptographic techniques, ADS-B broadcasts can be authenticated, secured from message modification and replay, and impersonation attacks. However, such solutions require the costly redesign of the ADS-B standard and the worldwide deployment of a security infrastructure. The cost and security challenges associated with key management and inter-operability outweigh the potential benefits [123].

Krozel et al. [136] have proposed to use a suite of Kalman filters to reduce noise within measured ADS-B signals. Noise reduction is intended to identify the wrong data and reduce the effect of data dropouts. Further, the authors have proposed integrity check mechanisms for ADS-B data using intent and geometric conformance. Intent conformance is the process by which the motion of an aircraft is compared with the broadcasted intent in vertical, horizontal, and speed dimensions. On the other hand, geometric conformance verifies that the aircraft state lies within the vertical and horizontal Required Navigation Performance (RNP) limits. For intent verification, they have proposed a correlation function using the information included in ADS-B signals. The aircraft state variables are verified independently by separate uncoupled Kalman filters.

6.6.1 Chapter Summary

We addressed the problem of the verifying the integrity of ADS-B navigation information without modifying the ADS-B standard. We proposed a PHY-layer verification method that exploits the Doppler spread phenomenon and the short coherence time of the channel between a prover aircraft and verifier aircraft to verify the velocity claims of the prover. The solution proposed in this work can be applied independently of the ADS-B standard. We further related the velocity claims to location claims through simple kinematic equations. We analyzed the security of our verification scheme and showed that it is equivalent to solving underdefined quadratic equation systems which are known to be hard.

This work can be extended to study the security and accuracy of the proposed method in different adversarial scenarios. A natural extension considers the collusion of multiple ground stations which coordinate their falsified signals to spoof a ghost aircraft. Intuitively, the fundamental problem of the adversary is that he is unable to solve the set of underdefined quadratic equations for determining the signals that need to be transmitted from the multiple ground stations. A more advanced (and costly) adversary model can consider an airborne attacker that spoofs ADS-B signals. The set of candidate trajectories that can be emulated by an airborne attacker warrant further investigation. Finally, the present work considers a verification process that occurs at cruising altitude and at cruising speed. The verification of ADS-B signals during other flight phases, such as takeoff and landing, requires further investigation.

CHAPTER 7

SECURE PHYSICAL LAYER VOTING

7.1 Introduction

7.1.1 Motivation

Distributed wireless networks fundamentally rely on the principle of cooperation. Nodes often share information to coordinate network functions and improve the fault-tolerance of distributed operations. As an example, cooperative spectrum sensing is known to improve the detection of licensed user activity in dynamic spectrum access (DSA) [44]. Data fusion is also widely used in wireless sensor networks (WSNs) for improving the performance of target detection, target tracking, and distributed sensing [45].

For many cooperative functions, binary voting algorithms increase fault-tolerance at relative low cooperation overhead. In binary voting, a community of distributed entities shares binary decisions ("yes" or "no") on a parameter of interest (e.g., channel state, presence of a target). A combining decision rule is applied to collectively determine the decision outcome. This rule is based on some form of majority voting, plurality or threshold, to achieve the desired level of reliability. Typically, binary votes are casted using a messaging scheme, in which 1-bit votes are carried by individual messages. However, message-based voting incurs relatively high voting delay. In this work, *we define the voting delay as the time period between the initiation of the voting process with the transmission of the first vote by any of the participants, until all votes have been received at the tallier.* The tallying time is not accounted as part of the voting delay. For message-based voting, each 1-bit vote is carried by a packet that contains PHY layer and a MAC layer headers. Moreover,

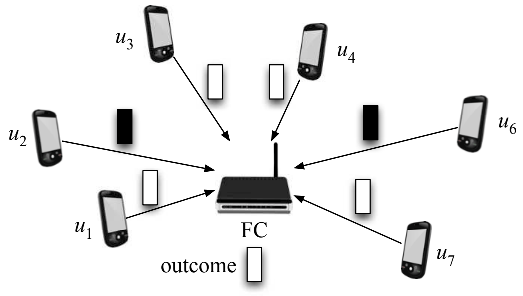


Figure 7.1: The PHYVOS voting scheme.

verifying the voter authenticity and protecting the integrity of binary votes via digital signatures and message authentication codes, requires additional packet fields. All additional fields (headers, message authentication codes, digital signature) increase the overall transmission time per vote. Further, voters must sequentially access the shared wireless channel to cast their votes. Most popular channel access protocols include anti-collision mechanisms (e.g., backoff process) that further increase the voting delay to cast multiple votes. For time-critical applications, a high voting delay could be unacceptable [46, 47].

As an example, consider the cooperative spectrum sensing mechanism proposed for DSA networks [44]. To accurately determine spectrum opportunities, secondary users sense licensed channels and submit state information ("busy" or "idle") to a fusion center (FC). The FC applies a combining decision rule (e.g., majority voting) to reliably determine the state of each channel. Existing federal regulations mandate that channel sensing must occur *every two seconds* [46], which leads to the frequent repetition of the fusion process. At such frequency, the time delay of message-based voting becomes problematic as the number of participants increases. Similar time and scalability constraints are encountered in control applications of networked multi-agent systems, where the consensus time requirement could be even more stringent [47].

To address the poor delay scalability of message-based voting, we present a *secure* and *fast* voting scheme called PHYVOS that implements voting at the PHY layer.

The basic principle of PHYVOS is shown in Figure 7.1. Wireless devices exploit the subcarrier orthogonality in the widely adopted orthogonal frequency division modulation (OFDM), to simultaneously cast their votes to an FC within just a few symbols. PHYVOS yields two distinct advantages relative to message-based voting. First, participants do not have to sequentially access the shared channel to cast their votes. This feature leads to significant delay savings, as delays due to contention and sequential access are eliminated. Second, votes do not carry long headers and cryptographic signatures that prolong the message transmission time. Therefore, PHYVOS drastically reduces the delay of voting, while maintaining a high security level. Implementing secure voting at the PHY layer involves new security and implementation challenges.

- Voting at the PHY layer is susceptible to false vote insertion and vote modification attacks, similar to message-based voting. An adversary can alter the voting outcome by exploiting the open nature of the wireless medium and manipulating the transmitted signals at the PHY layer. Without access to cryptographic primitives such as digital signatures and message authentication codes, securing the voting process is particularly challenging.
- The superposition of simultaneous transmissions from spatially-separated senders (voters) to a combined OFDM signal requires intricate transmitter and receiver designs [137, 138]. Senders must be synchronized in frequency and time to achieve symbol alignment at the receiver. Maintaining accurate synchronization in distributed systems could incur prohibitive coordination overheads [137].

7.1.2 Main Contributions and Chapter Organization

e design PHYVOS, a PHY-layer voting scheme that reduces the voting delay by several orders of magnitude compared to message-based voting. In PHYVOS, the voting delay, defined as the time required to cast votes, is reduced by exploiting the subcarrier orthogonality of OFDM to simultaneously cast votes from multiple participants. Vote tallying is performed at an FC that receives multiple votes as a single OFDM symbol. We further present a fully distributed version that allows every participant compute the vote tally, without the assistance of an FC. To overcome the challenges related to decoding simultaneous transmissions from multiple senders, binary votes are casted by adding energy to designated subcarriers. No transmission of preambles and headers is required, as the receiver does not demodulate the OFDM signal. Simple energy detection suffices. Moreover, relying on energy detection rather than message decodability for vote casting strengthens the security of our scheme, as it is generally hard to "erase" energy from a channel [14, 33].

We study the robustness of PHYVOS against an external and an internal adversary. The former attempts to modify votes by inserting energy into various subcarriers without knowing the subcarrier allocation. The latter is aware of any group secrets used to assign subcarriers, but not of pairwise secrets. PHYVOS guarantees the integrity of the voting outcome. We show that an active adversary who attempts to modify the casted votes, cannot flip the voting outcome at the FC with overwhelming probability. Also, the adversary cannot inject additional votes at the FC. We improve voting robustness by incorporating the transmission of multiple OFDM symbols to cast a single vote, thus realizing a repetition code. Since OFDM symbols have very short duration, a repetition code is still far more efficient than messaging. We analytically evaluate the voting robustness as a function of the relevant system parameters under a secret and an open vote model. We discuss practical implementation challenges of PHYVOS related to frequency and time synchronization. We present a prototype implementation of PHYVOS on the NI USRP platform. We complement the implementation with larger scale simulations and demonstrate the PHYVOS robustness to external and internal attacks.

PHYVOS is compatible with any wireless standard that is based on OFDM. This includes 802.11a/g/n/ac, WiMAX, UWB, DVB, and others. PHYVOS requires no hardware modifications of the OFDM TX/RX circuitry. Participants cast votes by transmitting regular OFDM symbols, and the RX can decipher votes at the FFT module of the OFDM receiver.

Chapter Organization: The remainder of this chapter is organized as follows. In Section 7.2, we present the system, communication, and adversary models. Section 7.3 describes PHYVOS. In Section 7.4, we analyze the security of PHYVOS under internal and external adversaries. A fully distributed version of PHYVOS without an FC is presented in Section 7.5. In Section, 7.6, we compare the overhead of PHYVOS with the overhead of message-based voting. Practical considerations and experimental verification of PHYVOS' performance are presented in Section 7.7. In Section 7.8, we discuss related work and conclude in Section 7.8.1.

7.2 Model Assumptions

7.2.1 Entities

The following entities are involved in the voting process:

- The *administrator* (A) is responsible for initializing the participants and the tallier with relevant cryptographic quantities, after verifying their identities.
- The M *participants* $u_1, u_2, \ldots, u_M$ cast M votes $v_1, v_2, \ldots, v_M$ to the tallier. Each vote reflects a binary choice.
- The *tallier* (R) is responsible for verifying and tallying the votes of all the participants by computing the voting outcome ($\mathcal{T}$).
- The *adversary* attempts to alter the voting outcome by injecting his own signals during the voting process.

In most applications, A and R could be the same entity such as the fusion center shown in Figure 7.1.

7.2.2 Voting Model

During the voting process, M participants cast M votes $v_1, v_2, \ldots, v_M$ to the tallier. For ease of illustration, we analyze the case where binary votes are casted, i.e., $v_i \in \{0, 1\}$,$\forall i$. The tallier R computes the voting outcome according to a threshold decision rule.

$$\mathcal{T} = \begin{cases} 1, & if \quad \sum_{i=1}^{M} (-1)^{v_i} \ < \ \gamma \\ 0, & if \quad \sum_{i=1}^{M} (-1)^{v_i} \ \geq \ \gamma. \end{cases} \tag{7.1}$$

The value of γ is application-dependent. As an example, by setting $\gamma = 0$, a plurality rule is implemented. Other values of γ allow for more relaxed or stricter agreement. The voting process must satisfy the requirements of correctness and robustness defined as follows.

Definition 5 (Correctness)**.** In the absence of attacks, all votes must be unambiguously recorded and tallied. That is, the voting must be error-free.

Definition 6 (Robustness)**.** A voting scheme is said to be robust against active attacks and faults, if the estimated outcome $\hat{\mathcal{T}}$ at the tallier equals the true outcome $\mathcal{T}$ computed by tallying the vote intend of all participants.

Robustness is a weaker requirement than accuracy, because it can be satisfied even if some votes are incorrectly tallied. However, robustness is sufficient for the intended applications of PHY-layer voting. We emphasize that other well-known voting requirements such as receipt-freeness [139], are beyond the scope of the envisioned applications of PHY-layer voting.

7.2.3 Communication Model

OFDM priliminaries: Orthogonal Frequency Division Multiplexing (OFDM) is a multicarrier modulation method adopted by many contemporary wireless technologies (e.g., 802.11a/g/n/ac/ad, LTE, WiMax, DVB-T) due to its high spectral efficiency. The main idea of OFDM is to divide the data stream to substreams, which are independently modulated in closely separated, orthogonal subcarriers. A basic block diagram of an OFDM system is shown in Figure 7.2. The data stream is fed to a serial-to-parallel (s2p) converter to generate N bit streams, where N is the number of available subcarriers for data transmission. The N streams are modulated (using BPSK, QPSK, QAM, etc.) and an N-point inverse Fourier transform (IFFT) is applied on the complex symbols. The IFFT output is fed to a parallel-to-serial (p2s) converter and further processed by a D/A converter to compose the baseband OFDM signal. At the receiver, after the downconversion to the baseband frequency, the analog signal is digitized by the A/D converter. The Fourier transform is applied to recover the complex constellation symbols and the N substreams are combined by a p2s converter to form the original data stream. The discrete time domain representation of the baseband OFDM signal $x(n)$ is given by [140]:

$$x(n) = \sum_{k=0}^{N-1} x_k(n) * e^{\frac{j2\pi nk}{N}}, \tag{7.2}$$

where $x_k(n) \in \{\alpha_1, \alpha_2, \ldots, \alpha_q\}$ is the complex modulated symbol at each of the N subcarriers transmitted at time n, and $\alpha_1, \alpha_2, \ldots, \alpha_q$ are the possible modulation symbol values (q denotes the modulation order). By selecting $x_k(n)$, the energy that is injected at each of the N subcarriers can be controlled. This energy is detected at an OFDM receiver by passing the time domain signal through an FFT. The energy detection at each subcarrier is the basic PHY-layer function exploited by PHYVOS for implementing the voting process.

We consider a one-hop communication topology, where every participant is either within the communication range of the tallier (star topology), or within one hop of each other (complete graph). Therefore votes are directly casted without a relay. Participants cast their votes to the tallier using an OFDM system with N orthogonal subcarriers, denoted by $f_1, f_2, \ldots f_N$. Participants could be at varying distances from the tallier. Moreover, participants and the tallier are synchronized to a time-slotted

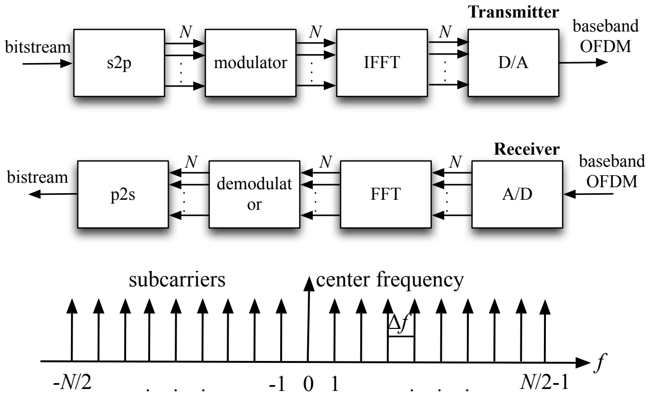


Figure 7.2: Block diagram of OFDM.

system with a maximum synchronization error of Δt, which depends on clock drifts and multipath. Note that time synchronization is already required for other network functions such as media access control.

Each participant must meet a minimum SNR requirement to cast a vote. This assumption is also true for message-based voting, where a sufficiently high SNR must be achieved to perform error-free decoding. As our method relies on energy detection, no other requirements are placed on the channel model. Different channels (e.g., AWGN, Rayleigh, Rician) could model the participant-to-tallier communications. If a participant's channel has an SNR below the required threshold for vote detection due to destructive interference, for all practical purposes this participant is no longer part of the voting. Finally, the channel state is assumed to be difficult to predict without being very close (within a few wavelengths) of the receiver, and without the transmission of preambles. This is true for most multipath scenarios, as it has been demonstrated by several works (e.g., [14, 33, 141, 142]).

7.2.4 Adversary Model

The adversary aims at flipping the voting outcome $\hat{\mathcal{T}}$ computed at the tallier. The adversary could be *external* or *internal*. An external adversary is unaware of any

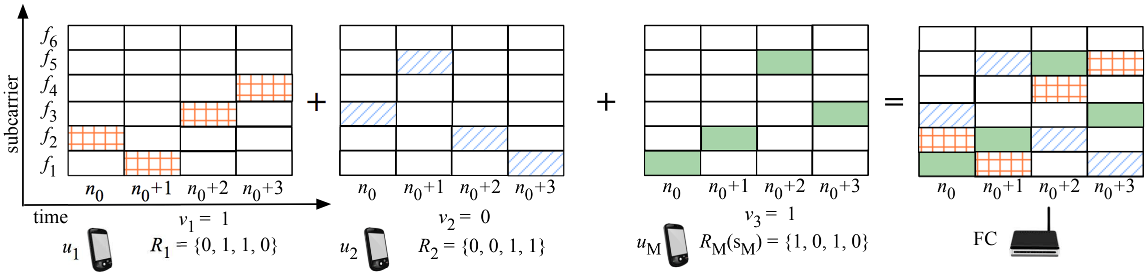


Figure 7.3: The vote casting phase for M participants voting over N subcarriers (here $N = 2M$).

cryptographic primitives used to initialize participants. An internal adversary on the other hand, is a legitimate participant with access to any group secrets. We assume that the adversary does not launch denial-of-service (DoS) attacks that prevent the computation of any voting outcome (e.g. by eliminating the votes of every participant). Such an attack is easily detectable. The adversary is loosely synchronized to the tallier with the same synchronization error as the rest of the participants. Two different voting models are considered with respect to the secrecy of the vote intent of each participant:

Secret vote model: In the secret vote model, the adversary is not aware of the vote intent of the participants.

Open vote model: In this model, the adversary is aware of the vote intent of the targeted participants. The vote intent can be determined by some side-channel information. For instance, in a spectrum sensing application for CRNs, the vote intent of an honest participant can be determined by performing spectrum sensing on a nearby location.

7.3 PHYVOS: Physical Layer Voting

The key principle of PHYVOS is to simultaneously cast votes by injecting energy on designated subcarriers. An adversary attempting to modify a vote on subcarrier

f_i, would have to "erase" the signal received by the tallier on f_i and simultaneously inject energy on some other subcarrier. This is generally a hard problem that requires knowledge of the signal transmitted at f_i, the precise time that the signal was transmitted, the signal propagation delay, and precise channel state information [14, 33, 143]. This knowledge needs to be collected and synchronized for all voters. PHYVOS consists of four phases: the setup phase, the vote request phase, the vote casting phase, and the tallying phase.

7.3.1 Setup Phase

In the setup phase, the administrator initializes the tallier and the M participants. If the tallier and the administrator are the same entity, only the M participants need to be initialized. The initialization process is as follows.

Key generation: The administrator $\mathcal{A}$ executes a probabilistic key generator algorithm $KeyGen(1^\tau) \rightarrow K$. This algorithm takes as input a security parameter τ, and outputs a master key K. $\mathcal{A}$ derives $M+1$ additional keys from K with $K_{perm} = H_{perm}(K)$ and $K_{vote,i} = H_{vote}(K, i)$ for $i = [1..M]$, where H_{perm} and H_{vote} are cryptographic hash functions.

Key assignment: $\mathcal{A}$ loads K_{perm} and $K_{vote,i}$ with $i \in [1..M]$ to R. It also loads K_{perm} and $K_{vote,i}$ to each u_i. At the end of the setup phase, R shares one pairwise key $K_{vote,i}$ with each u_i and a common key K_{perm} with all u_is.

7.3.2 Vote Request Phase

In the vote request phase, the tallier synchronizes all participants for simultaneous voting. This phase is necessary to ensure that delay overhead gains are achieved by the simultaneous vote casting. Periodic or on-demand voting can be employed to request a vote. In periodic voting, participants exploit their synchronization to a common time-slotted system to cast their votes at fixed time intervals without an explicit request from the tallier. This operation mode is suitable for periodic network operations. In on-demand voting, the tallier broadcasts a vote request synchronization message to all participants to initiate the voting process.

7.3.3 Vote Casting Phase

During the vote casting phase, participants simultaneously cast their votes to the tallier. Each vote v_i consists of a series of ℓ symbol votes $v_i(n_0), v_i(n_0{+}1), \ldots, v_i(n_0{+}\ell{-}1)$ casted over ℓ consecutive time slots. The ℓ symbol votes operate as a repetition code to improve the robustness of vote casting in the presence of an adversary. To cast a symbol vote $v_i(n)$ at the n^{th} time slot, a participant u_i is assigned two subcarriers $f^0_{u_i}(n)$ and $f^1_{u_i}(n)$. One subcarrier is used to cast a "no" vote whereas the other is used to cast a "yes" vote. We note that in the absence of an adversary, a single subcarrier is sufficient to cast a binary vote. However, the adversary could easily modify the vote that corresponds to energy absence by injecting energy on the alternative subcarrier. Therefore, we adopt a two-subcarrier solution.

Moreover, the subcarriers assigned to each participant are permuted per time slot to hide the assignment from the adversary. This is achieved by applying a pseudo-random permutation on the subcarrier assignment. Finally, for a given assignment $f^0_{u_i}(n)$ and $f^1_{u_i}(n)$ to participant u_i, the mapping to "yes" and "no" votes is randomized by the application of a pseudo-random binary sequence shared between u_i and R. This prevents an internal adversary from determining the subcarrier that corresponds to a specific vote. Formally, vote casting involves the following steps:

1. **Subcarrier assignment:** Each participant u_i applies pseudo-random function

$$\Pi_F : \{0,1\}^\tau \times [1..N] \times \mathbb{Z}^+ \rightarrow [1..N],$$

to map subcarrier with index p during slot n, to subcarrier $\Pi_F(K_{perm}, p, n)$.

Participant u_i is assigned subcarriers

$$\begin{aligned} f^0_{u_i}(n) &= f_{\Pi_F(K_{perm},(2i-1),n)}, \\ f^1_{u_i}(n) &= f_{\Pi_F(K_{perm},2i,n)}. \end{aligned}$$

2. **Pseudo-random sequence generation:** Each participant u_i applies pseudo-random generator function

$$\Phi : \{0,1\}^\tau \times \mathbb{Z}^+ \to \{0,1\}$$

to generate a binary sequence $R_i = \{r_i(1), r_i(2), \ldots\}$ with $r_i(n) = \Phi(K_{vote,i}, n)$.

3. **Symbol vote casting:** Let voting casting be initiated at slot n_0. To cast a vote $v_i \in \{0,1\}$, a participant u_i generates ℓ symbol votes $v_i(n_0) = v_i(n_0{+}1) = \ldots = v_i(n_0 + \ell - 1) = v_i$. Each $v_i(n)$ is represented by an OFDM symbol with the following values per subcarrier

$$x_k(n) = \begin{cases} \alpha_y, & f^{v_i(n) \oplus r_i(n)}_{u_i}(n) \\ 0, & \text{otherwise,} \end{cases} \tag{7.3}$$

where α_y is a randomly selected modulation symbol and $n_0 \leq n < n_0 + \ell$. Note that the placement of energy of either $f^0_{u_i}(n)$ or $f^1_{u_i}(n)$ is based on the XOR between the vote value $v_i(n)$ and the random bit $r_i(n)$.

The vote casting phase for three participants and six subcarriers is shown in Figure 7.3. In Step 1, participants apply the pseudo-random permutation to obtain the subcarrier assignment. For the first four time slots, the subcarrier permutations are $\{f_6,f_2,f_3,f_4,f_1,f_5\}$, $\{f_1,f_3,f_5,f_6,f_4,f_2\}$, $\{f_3,f_1,f_6,f_2,f_5,f_4\}$ and $\{f_5,f_4,f_2,f_1,f_6,f_3\}$. Participant u_1 is assigned $\{f^0_{u_1}, f^1_{u_1}\}$: $\{(f_6,f_2),(f_1,f_3),(f_3,f_1),(f_5,f_4)\}$, participant u_2 is assigned $\{(f_3,f_4),(f_5,f_6),(f_6,f_2),(f_2,f_1)\}$ and participant u_3 is assigned $\{(f_1,f_5),(f_4,f_2),(f_5,f_4),(f_6,f_3)\}$. In Step 2, each participant u_i generates the pseudo-random sequence for slots $1 - 4$. For u_1, $r_1 = \{0, 1, 1, 0\}$, for u_2, $r_2 = \{0, 0, 1, 1\}$,, and for for u_3, $r_3 = \{1, 0, 1, 0\}$,. In Step 3, participants cast votes at the designated subcarriers. In our example, u_1 wants to cast a "yes" vote ($v_1 = 1$). He XORs v_1 with r_1 and determines the active subcarriers as $\{f_2, f_1, f_3, f_4\}$ for slots $1, 2, 3$, and 4, respectively. The active subcarriers for other participants are similarly determined. The symbol votes arrive (almost) time aligned at the tallier such that OFDM symbols are formed as shown in Figure 7.3.

7.3.4 Vote Tallying Phase

In the vote tallying phase, the tallier computes the voting outcome $\mathcal{T}$ according to the threshold rule in (7.1). To infer the votes of each participant, the tallier computes the FFT of the digitized baseband OFDM signal to separate the spectral components to each of the subcarriers. The tallier then uses an energy detector at each output of the FFT block to detect the transmitted symbol votes. Note here that *no symbol demodulation is necessary to determine the presence of energy.* At time n, a symbol vote $v_i(n)$ is computed only if the detected average power is beyond a threshold γ_D on only one of the two designated subcarriers. In any other case, the symbol vote is recorded in error. Formally, for a participant u_i, the recovery of v_i at the tallier is performed as follows.

1. **Energy detection:** Sample the FFT output of subcarriers $f^0_{u_i}(n)$ and $f^1_{u_i}(n)$ assigned to u_i and compute the average received power over L samples:

$$p^0_{u_i}(n) = \frac{1}{L}\sum_{i=1}^{L} |y_j(i)|^2, \quad p^1_{u_i}(n) = \frac{1}{L}\sum_{i=1}^{L} |y_{j+1}(i)|^2, \tag{7.4}$$

   with $n_0 \leq n < n_0 + \ell$.

2. **Extract symbol votes:** The symbol votes $\hat{v}_i(n)$ are computed by XORing the subcarrier superscript were energy was detected with the pseudo-random sequence shared between u_i and the tallier to correctly map the subcarrier

index to the vote value.

$$\hat{v}_i(n) = \begin{cases} 0 \oplus r_i(n), & \text{if } p^0_{u_i}(n) > \gamma_D, \quad p^1_{u_i}(n) \leq \gamma_D \\ 1 \oplus r_i(n), & \text{if } p^0_{u_i}(n) \leq \gamma_D, \quad p^1_{u_i}(n) > \gamma_D \\ e, & \text{otherwise.} \end{cases} \tag{7.5}$$

with $n_0 \leq n < n_0 + \ell$.

3. **Compute the final vote:** The final vote $\hat{v}_i$ is computed by discarding all inconclusive symbol votes.

$$\hat{v}_i = \begin{cases} 0, & \text{if } \sum_{n=n_0, v_i(n) \neq e}^{n_0+\ell-1} (-1)^{v_i(n)} > 0 \\ 1, & \text{if } \sum_{n=n_0, v_i(n) \neq e}^{n_0+\ell-1} (-1)^{v_i(n)} < 0 \\ e, & \text{otherwise.} \end{cases} \tag{7.6}$$

4. **Compute the final voting outcome:** The final voting outcome $\hat{\mathcal{T}}$ computed according to:

$$\hat{\mathcal{T}} = \begin{cases} 1, & if \quad \sum_{i=1}^{M} (-1)^{\hat{v}_i} \; < \; \gamma \\ 0, & if \quad \sum_{i=1}^{M} (-1)^{\hat{v}_i} \; \geq \; \gamma. \end{cases} \tag{7.7}$$

The voting outcome $\hat{\mathcal{T}}$ is estimated by tallying all votes using eq. (7.1), where the vote values v_i have been substituted by their estimates $\hat{v}_i, i \in [1..M]$. The tallying operation is shown in the example of Figure 7.3. For participant u_1, the tallier detects an average power over γ_D on subcarriers $\{f_2, f_1, f_3, f_4\}$. By XORing the output $\{1, 0, 0, 1\}$ with the random sequence $R_1 = \{0, 1, 1, 0\}$, it obtains the symbol votes $\hat{v}_1(n_0) = 1$, $\hat{v}_1(n_0 + 1) = 1$, $\hat{v}_1(n_0 + 2) = 1$, and $\hat{v}_1(n_0 + 3) = 1$, indicating a final vote $\hat{v}_1 = 1$. Similarly, participant u_2 uses random sequence $R_2 = \{1, 1, 0, 1\}$ to compute $v_2 = 0$. The vote computation proceeds in parallel for all participants. The voting outcome is estimated to be $\hat{\mathcal{T}} = 1$.

7.4 Security Analysis

In this section, we evaluate the robustness of PHYVOS under the external and internal adversary model.

7.4.1 External Adversary

Under the external adversary model, the adversary is unaware of the cryptographic keys K_{perm} and $K_{vote,i}$ used to permute the subcarrier assignment per time slot and also randomize the symbol votes. Therefore, his best strategy is to inject energy on randomly selected subcarriers. Let us consider the vote v_i of u_i, consisting of ℓ symbol votes $v_i(n_0)$, $v_i(n_0+1), \ldots, v_i(n_0+\ell-1)$,. To successfully cast v_i, the adversary must guess the subcarrier of u_i that dictates the vote opposite to v_i for ℓ symbol votes. Even if the subcarrier is correctly guessed, the adversary cannot "erase" the energy injected by the legitimate participant on the complementary subcarrier. Erasure of the modulation symbol a_y transmitted by u_i requires the a priori knowledge of a_y, knowledge of the channel between the voter and the tallier as well as the adversary and the tallier, and precise synchronization between the voter and the adversary [143]. We note that u_i randomly selects a_y for each symbol vote. Moreover, the channel between u_i and tallier rapidly decorrelates with the distance from u_i. Unless the adversary is very close to u_i, the channel between u_i and tallier is unpredictable [144].

Without the opportunity to flip votes, the adversary can flip the voting outcome if he nullifies a sufficient number of in favor votes to overcome the decision threshold γ. Vote nullification occurs, if energy is present on both subcarriers assigned to a participants over the ℓ symbol votes. Let the adversary inject energy on $J \leq N$ subcarriers of his choice in order to flip the voting outcome $\mathcal{T}$. Without loss of generality assume that votes in favor of $\mathcal{T}$ outnumber the votes against $\mathcal{T}$ by a voting margin μ. The probability of flipping the outcome to an estimate $\hat{\mathcal{T}} \neq \mathcal{T}$ is given in Proposition 10.

Proposition 10. *Let participants cast M votes over $2M \leq N$ subcarriers by transmitting ℓ symbol votes. Let an external adversary inject energy on $J \leq N$ subcarriers. The probability of flipping the voting outcome is*

$$\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] = \sum_{z=\mu-\gamma}^{\min\{n_1,J\}} \sum_{x=z}^{\min\{n_1,J\}} \Pr[\mathbf{Z} = z], \tag{7.8}$$

where $n_1 = \frac{M+\mu}{2}$ denotes the number of votes in favor of $\mathcal{T}$ and $n_2 = \frac{M-\mu}{2}$ denotes the votes against $\mathcal{T}$.

$$\begin{aligned} \Pr[\mathbf{Z} = z] \quad = \quad & \sum_x \Bigg(\binom{n_1}{x} \binom{n_2}{x-z} \left(\frac{\binom{N-2x+z}{J-2x+z}}{\binom{N}{J}} \right)^{\ell} \\ & - \Bigg(\sum_{w=1}^{\min\{n_1-x,J-x\}} \sum_{k=0}^{w-z} \binom{n_1-x}{w} \\ & \binom{n_2-x+z}{k} \frac{\binom{N-n_1-n_2-4x+2z}{J-w-k-2x+z}}{\binom{N-x}{J-x}} \Bigg) \\ & \left(\frac{1}{\binom{N}{J}} \right)^{\ell} \Bigg). \end{aligned} \tag{7.9}$$

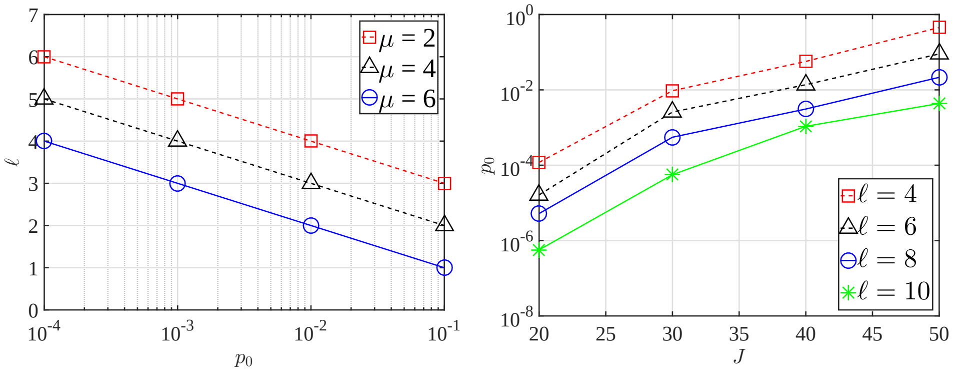
Proof. Let a vote process with M participants lead to a voting outcome $\mathcal{T}$, selected with a margin μ. Without loss of generality, assume that the votes in favor of $\mathcal{T}$ are "yes" votes. For a margin μ, it is straightforward to show that there are $n_1 = \frac{M+\mu}{2}$ "yes" votes and $n_2 = \frac{M-\mu}{2}$ "no" votes. To flip $\mathcal{T}$ through vote nullification, the adversary must nullify at least $\mu - \gamma$ more "yes" votes than "no" votes to make the vote difference less or equal to γ.

Let $\mathbf{X}$ be a random variable (RV) denoting number of nullified "yes" votes when the adversary injects energy on J subcarriers on each of the ℓ voting slots. To nullify x "yes" votes, the adversary has to pick at each slot those subcarriers that nullify the "yes" votes of a set of x participants. Similarly, let $\mathbf{Y}$ be an RV denoting the number of nullified "no" votes, when the adversary injects energy on J subcarriers on each of the ℓ voting slots. Let also $\mathbf{Z} = \mathbf{X} - \mathbf{Y}$ denote the excess number of nullified "yes" votes relative to "no" votes. The pmf of $\mathbf{Z}$ can be computed using

$$
\begin{aligned}
\Pr[\mathbf{Z} = z] &= \sum_{x} \Pr[\mathbf{X} = x, \mathbf{Y} = x - z] \\
&= \sum_{x} \left(\binom{n_1}{x} \binom{n_2}{x - z} \left(\frac{\binom{N-2x+z}{J-2x+z}}{\binom{N}{J}} \right)^{\ell} \right. \\
&\quad - \left(\sum_{w=1}^{\min\{n_1-x, J-x\}} \sum_{k=0}^{w-z} \binom{n_1 - x}{w} \binom{n_2 - x + z}{k} \frac{\binom{N-n_1-n_2-4x+2z}{J-w-k-2x+z}}{\binom{N-x}{J-x}} \right) \\
&\qquad \left. \left(\frac{1}{\binom{N}{J}} \right)^{\ell} \right)
\end{aligned}
\tag{7.10}
$$

In eq. (7.10), the first term denotes all possible combinations of x subcarriers that nullify x out of n_1 "yes" votes, which fixes the combination of x votes that are nullified after ℓ slots. Similarly, it has all possible combinations of $x - z$ subcarriers that nullify $x - z$ out of n_2 "no" votes. This term is multiplied by the number of ways of choosing $J - 2x + z$ subcarriers from the remaining $N - 2x + z$, and divided by all possible ways of choosing J subcarriers out of N. The second multiplier is raised to the power of ℓ because the subcarrier selection is repeated with every time slot in an independent fashion. Note that the first multiplier is not raised to the power of ℓ because the set of votes to be nullified remains fixed after the first slot. The second term, excludes all possible combinations of additional "yes" votes and "no" being nullified due to the selection of the remaining $J - 2x + z$ subcarriers. This term computes all possible selections of $J - 2x + z$ subcarriers in which at least one subcarrier is assigned to the remaining $n_1 - x$ "yes" votes and $n_2 - x + z$ "no" votes and this subcarrier is present on all the ℓ slots, multiplied by the probability of occurrence for each selection.

The probability of flipping the voting outcome is equal to the probability of nullifying at least $\mu - \gamma$ more "yes" than "no" votes, i.e., $\mathbf{Z} \geq \mu - \gamma$. Summing



(a) (b)

Figure 7.4: (a) Minimum number of symbol votes ℓ to guarantee robustness p_0 for an external adversary for va, (b) minimum number of symbol votes ℓ to guarantee robustness p_0 for $\mu = 4$ and various J.

(7.10) over all $z \geq \mu - \gamma$ yields,

$$\begin{aligned}\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] &= Pr[\mathbf{Z} \geq \mu - \gamma] \\ &= \sum_{z=\mu-\gamma}^{\min\{n_1,J\}} \sum_{x=z}^{\min\{n_1,J\}} \Pr[\mathbf{Z} = z], \qquad (7.11)\end{aligned}$$

where $\Pr[\mathbf{Z} = z]$ is given by (7.10). This completes the proof. □

Selecting the Security Parameter ℓ: Proposition 10 allows us to select the number of symbol votes ℓ to guarantee robustness with a desired probability p_0. The following corollary yields a lower bound on ℓ such that $\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] \leq p_0$.

Corollary 2. *For an external adversary,* $\Pr\left[\hat{\mathcal{T}} \neq \mathcal{T}\right] \leq p_0$ *if*

$$\ell > \lceil \frac{1}{\log \frac{1}{C_1}} \log \frac{C_0}{p_0} \rceil, \qquad (7.12)$$

where

$$C_0 = \sum_{z=\mu-\gamma}^{\min\{n_1,J\}} \sum_{x=z}^{\min\{n_1,J\}} \left(\binom{n_1}{x} \binom{n_2}{x-z} \right), \qquad (7.13)$$

$$C_1 = \sum_{z=\mu-\gamma}^{\min\{n_1,J\}} \sum_{x=z}^{\min\{n_1,J\}} \frac{\binom{N-2x+z}{J-2x+z}}{\binom{N}{J}}. \tag{7.14}$$

Proof. We wish to determine the value of ℓ for which $\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] \leq p_0$. From (7.11), it follows that

$$\begin{aligned}
\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] &= \sum_{z=\mu-\gamma}^{\min\{n_1,J\}} \sum_{x=z}^{\min\{n_1,J\}} \left(\binom{n_1}{x} \binom{n_2}{x-z} \left(\frac{\binom{N-2x+z}{J-2x+z}}{\binom{N}{J}} \right)^{\ell} \right. \\
&\quad \left. - \left(\sum_{w=1}^{\min\{n_1-x,J-x\}} \sum_{k=0}^{w-z} \binom{n_1-x}{w} \binom{n_2-x+z}{k} \frac{\binom{N-n_1-n_2-4x+2z}{J-w-k-2x+z}}{\binom{N-x}{J-x}} \right) \left(\frac{1}{\binom{N}{J}} \right)^{\ell} \right) \\
&< \sum_{z=\mu-\gamma}^{\min\{n_1,J\}} \sum_{x=z}^{\min\{n_1,J\}} \left(\binom{n_1}{x} \binom{n_2}{x-z} \left(\frac{\binom{N-2x+z}{J-2x+z}}{\binom{N}{J}} \right)^{\ell} \right) \\
&< \sum_{z=\mu-\gamma}^{\min\{n_1,J\}} \sum_{x=z}^{\min\{n_1,J\}} \left(\binom{n_1}{x} \binom{n_2}{x-z} \right) \sum_{z=\mu-\gamma}^{\min\{n_1,J\}} \sum_{x=z}^{\min\{n_1,J\}} \left(\frac{\binom{N-2x+z}{J-2x+z}}{\binom{N}{J}} \right)^{\ell} \\
&< \sum_{z=\mu-\gamma}^{\min\{n_1,J\}} \sum_{x=z}^{\min\{n_1,J\}} \left(\binom{n_1}{x} \binom{n_2}{x-z} \right) \left(\sum_{z=\mu-\gamma}^{\min\{n_1,J\}} \sum_{x=z}^{\min\{n_1,J\}} \frac{\binom{N-2x+z}{J-2x+z}}{\binom{N}{J}} \right)^{\ell}
\end{aligned} \tag{7.15}$$

Limiting the right hand side of (7.15) by p_0 and solving for ℓ,

$$\ell > \lceil \frac{1}{\log \frac{1}{C_1}} \log \frac{C_0}{p_0} \rceil, \tag{7.16}$$

where

$$C_0 = \sum_{z=\mu-\gamma}^{\min\{n_1,J\}} \sum_{x=z}^{\min\{n_1,J\}} \left(\binom{n_1}{x} \binom{n_2}{x-z} \right), \tag{7.17}$$

$$C_1 = \sum_{z=\mu-\gamma}^{\min\{n_1,J\}} \sum_{x=z}^{\min\{n_1,J\}} \frac{\binom{N-2x+z}{J-2x+z}}{\binom{N}{J}}. \tag{7.18}$$

This completes the proof. □

From Corollary 2, we observe that the required number of symbol votes ℓ drops linearly with the logarithm of p_0. This is also confirmed by Figure 7.4(a), which shows ℓ as a function of p_0, for a total for 20 participants voting over 52 subcarriers. In Figure 7.4(a), we set $J = 12$, $\gamma = 0$ (plurality rule) and varied the vote margin μ. We observe that a relatively small number of symbol votes ($\ell < 5$) allows us to achieve high levels of robustness for relatively small margins.

An obvious tactic for the adversary is to increase the number of attacked subcarriers. In Figure 7.4(b), we show the number of symbol votes required to achieve a desired robustness level for various J, when the vote margin is fixed to $\mu = 4$. We observe that if small number of subcarriers are attacked, the achieved robustness is high for small ℓ. The adversary's success increases with J at the expense of increased presence over the various subcarriers.

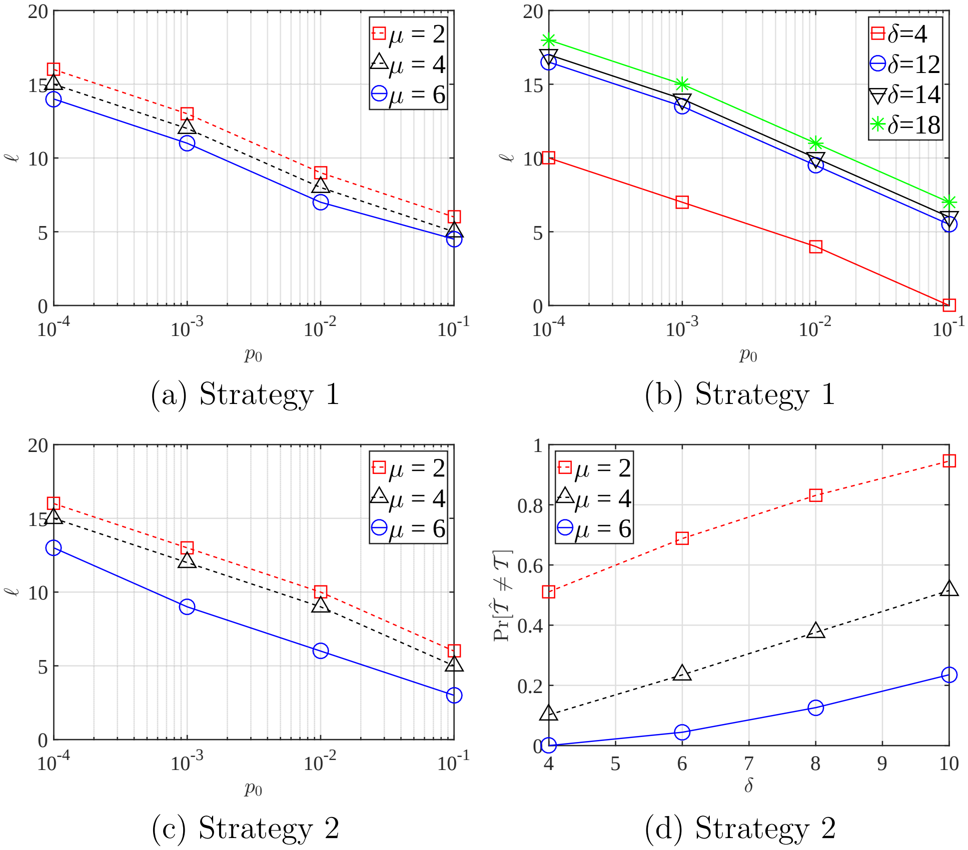
To prevent the adversary from flipping the voting outcome via vote nullification, the tallier can reject the voting outcome if the fraction of nullified votes exceeds a certain threshold. This threshold can be defined to exceed the expected number of nullified votes under unintentional interference. In Section 7.7, we explore this prevention method by determining the pmf for the number of nullified votes due to the imperfections of the wireless channel. The pmf is used to select the threshold for rejecting the voting outcome.

7.4.2 Internal Adversary

An internal adversary could be any malicious participant aiming at manipulating the voting outcome. Such an adversary has knowledge of the key K_{perm} used for the subcarrier assignment. Therefore, it can target particular subcarriers to nullify votes of certain participants. Note that we do not consider the case where the adversary compromises the credentials (pairwise keys) of several participants by, for example, gaining access to the participants' devices. In this case, the adversary can impersonate the compromised participants and cast votes on their behalf. For all practical purposes, such impersonations cannot be authenticated using cryptographic methods, and can only be detected using radio fingerprinting methods. Such attacks are possible against message-based voting systems as well, and cannot be defended by standard cryptographic methods of authentication and message integrity.

Modifying a Single Vote: We first analyze the modification of vote v_i of a targeted participant u_i. Let u_i initiate its voting at time slot n_0 by submitting ℓ symbol votes. Although the adversary is aware of the subcarriers assigned to u_i, he is unaware of the pseudo-random sequences used to map the subcarriers to the "yes/no" votes. Without access to the pairwise key $K_{vote,i}$, the adversary can at best guess the subcarrier where energy must be injected to emulate a "yes" or a "no" vote. We consider two possible adversary strategies. In the first strategy, the adversary randomly selects one of u_i's subcarriers to emulate a target vote. In the second strategy, the adversary nullifies vote v_i by injecting energy on both subcarriers assigned to u_i.

Strategy 1: In the first strategy, the adversary A emulates the voter behavior by injecting energy to either $f^0_{u_i}$ or $f^1_{u_i}$. Let A target the casting of $v_i = 0$. To successfully cast v_i, he can guess the subcarrier mapping with success probability 0.5, for every symbol vote. The adversary can still hope to nullify the vote of u_i (i.e., change the value of $\hat{v}_i(n)$ from $\hat{v}_i(n) = v_i$ to $\hat{v}_i(n) = e$). According to (7.6), to nullify $\hat{v}(i)$, all symbol votes $\hat{v}_i(n_0), \hat{v}_i(n_0+1), \ldots, \hat{v}_i(n+0+\ell-1)$ must



(a) Strategy 1

(b) Strategy 1

(c) Strategy 2

(d) Strategy 2

Figure 7.5: Minimum number of symbol votes ℓ to guarantee robustness p_0 (a) under the secret vote model for Strategy 1, (b) for $\mu = 4$ and various δ, under the secret vote model and for Strategy 1, (c) under the open vote model for Strategy 2, (d) under the secret vote model for Strategy 2.

be nullified. This is equivalent to guessing the subcarrier index used by u_i to cast each of the ℓ symbol votes. As the subcarrier carrying each symbol vote is selected pseudo-randomly and independently per symbol vote, the probability of nullifying $\hat{v}_i$ becomes:

$$\begin{aligned} \Pr[\hat{v}(i) = e] &= \Pr[\hat{v}_i(n_0) = e, \ldots, \hat{v}_i(n_0 + \ell - 1) = e] \\ &= 0.5^{\ell}. \qquad (7.19) \end{aligned}$$

Note that eq. (7.19) is true even if the value of $\hat{v}_i$ is known a priori because the index of the subcarrier carrying $\hat{v}_i(n)$ is XORed with $r_i(n)$ (see eq. (7.3)). From (7.19), we can select ℓ to drive $\Pr[\hat{v}(i) = e]$ to any desired level.

Modifying the Voting Outcome: We now analyze the probability of modifying the voting outcome under the secret vote model and the open vote model stated in Section 7.2.4.

Proposition 11. *Let an internal adversary attempt to nullify the votes of δ participants and let $p = \Pr[v(i) = e]$ denote the probability of nullifying a singe vote, as given by* (7.19)*. Under the secret vote model, an internal adversary following Strategy 1 can flip the voting outcome for a decision threshold γ and a margin μ with probability*

$$\begin{aligned}\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] &= \sum_{i=\mu-\gamma}^{\delta} HG(n_1, M, i, \delta) \\ &\quad \sum_{z=\mu-\gamma}^{i} \sum_{x=z}^{\min\{i, \frac{\delta+z}{2}\}} \frac{\binom{i}{x}\binom{\delta-i}{x-z}}{\binom{\delta}{2x-z}} B(2x-z, \delta, p),\end{aligned}$$

where $n_1 = \frac{M+\mu}{2}$ denotes the number of votes in favor of $\mathcal{T}$.

Proof. Let a vote process with M participants lead to a voting outcome $\mathcal{T}$, selected with a margin μ. Without loss of generality, assume that the votes in favor of $\mathcal{T}$ are "yes" votes. For a margin μ, it is straightforward to show that there are $n_1 = \frac{M+\mu}{2}$ "yes" votes and $n_2 = \frac{M-\mu}{2}$ "no" votes. To flip $\mathcal{T}$ through vote nullification, the adversary must nullify at least $\mu - \gamma$ more "yes" votes than "no" votes to make the vote difference less or equal to γ. For an adversary that attempts to nullify a total of δ votes, the probability that i of them are "yes" votes is given by a hypergeometric distribution.

$$\Pr[\mathbf{I} = i] = HG(n_1, N, i, \delta) \tag{7.20}$$

Each vote is successfully nullified with probability $p = \Pr[v_i = e] = 0.5^{\ell}$. Let

$\mathbf{X}$ be an RV denoting the number of successfully nullified "yes" votes, when x "yes" votes are attacked. Because the nullification of each vote is an independent Bernoulli trial (the adversary randomly picks one of the two subcarriers assigned to each attacked participant), $\mathbf{X}$ follows the binomial distribution

$$\Pr[\mathbf{X} = x] = B(x, i, p), \quad p = 0.5^{\ell}. \tag{7.21}$$

Similarly, let $\mathbf{Y}$ be an RV denoting the number of "no" votes that are successfully nullified. For $\mathbf{Y}$,

$$\Pr[\mathbf{Y} = y] = B(y, \delta - i, p), \quad p = 0.5^{\ell}. \tag{7.22}$$

The probability that the number of successfully nullified "yes" votes exceeds the number of nullified "no" votes by exactly z votes is given by RV $\mathbf{Z} = \mathbf{X} - \mathbf{Y}$. The pmf of $\mathbf{Z}$ can be computed using the convolution formula.

$$\begin{aligned}
\Pr[\mathbf{Z} = z] &= \sum_x \Pr[\mathbf{X} = x, \mathbf{Y} = x - z] \\
&= \sum_x \Pr[\mathbf{X} = x] \Pr[\mathbf{Y} = x - z | \mathbf{X} = x] \\
&= \sum_x B(x, i, p) B(x - z, \delta - i, p) \\
&= \sum_{x=z}^{\min\{i, \frac{\delta+z}{2}\}} \frac{\binom{i}{x}\binom{\delta-i}{x-z}}{\binom{\delta}{2x-z}} B(2x - z, \delta, p).
\end{aligned} \tag{7.23}$$

The probability of flipping the voting outcome is equal to the probability of nullifying at least $\mu - \gamma$ more "yes" than "no" votes, i.e., $\mathbf{Z} \geq \mu - \gamma$. Summing (7.23) over all $z \geq \mu - \gamma$ yields,

$$Pr[\mathbf{Z} \geq \mu - \gamma] = \sum_{z=\mu-\gamma}^{i} \sum_{x=z}^{\min\{i, \frac{\delta+z}{2}\}} \frac{\binom{i}{x}\binom{\delta-i}{x-z}}{\binom{\delta}{2x-z}} B(2x - z, \delta, p). \tag{7.24}$$

Using (7.20) and (7.24), we compute

$$\begin{aligned}\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] &= \sum_{i=\mu-\gamma}^{n_1} \Pr[I=i] Pr[\mathbf{Z} \geq \mu-\gamma] \\ &= \sum_{i=\mu-\gamma}^{n_1} HG(n_1, N, i, \delta) \sum_{z=\mu-\gamma}^{i} \sum_{x=z}^{\min\{i, \frac{\delta+z}{2}\}} \frac{\binom{i}{x}\binom{\delta-i}{x-z}}{\binom{\delta}{2x-z}} B(2x-z, \delta, p).\end{aligned}$$

□

Proposition 12. *Under the open vote model, an internal adversary following Strategy 1 can flip the voting outcome for a decision threshold γ and a margin μ with probability*

$$\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] = \sum_{i=\mu-\gamma}^{\delta} B(i, \delta, p). \tag{7.25}$$

where δ denotes the number of votes that the adversary attempts to nullify, with $\delta \leq n_1$.

Proof. Let a vote process with M participants lead to a voting outcome $\mathcal{T}$, selected with a margin μ. Without loss of generality, assume that the votes in favor of $\mathcal{T}$ are "yes" votes. For a margin μ, it is straightforward to show that there are $n_1 = \frac{M+\mu}{2}$ "yes" votes and $n_2 = \frac{M-\mu}{2}$ "no" votes. Consider a voting outcome $\mathcal{T}$ with a margin μ with the in favor votes be "yes" votes. Under the open vote model, the adversary only targets subcarriers that are assigned to participants that intend to vote "yes". The voting outcome $\mathcal{T}$ is flipped if at least $\mu-\gamma$ "yes" votes are nullified. The adversary successfully nullifies an attacked vote with probability $p = 0.5^{\ell}$. As the success of nullifying each vote is an independent event (the adversary picks one of the two subcarriers assigned to each attacked participant at random), the number of nullified "yes" votes when a total of δ "yes" votes are attacked, follows the binomial distribution with parameter p.

$$\Pr[\mathbf{X} = x] = B(x, \delta, p), \quad p = 0.5^{\ell}. \tag{7.26}$$

Summing over all values of $x \geq \mu - \gamma$ yields,

$$\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] = \sum_{i=\mu-\gamma}^{\delta} B(i, \delta, p), \quad p = 0.5^{\ell}. \tag{7.27}$$

The value of δ is smaller or equal to the number n_1 of "yes" votes, as there is no benefit to nullifying "no" votes. □

Selecting the Security Parameter ℓ: Propositions 11 and 12 allow us to select the number of symbol votes ℓ to guarantee robustness with a desired probability. Suppose we want to limit $\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] \leq p_0$. Then, we can select ℓ to guarantee p_0, as shown in Corollaries 3 and 4.

Corollary 3. *For the secret vote model,* $\Pr\left[\hat{\mathcal{T}} \neq \mathcal{T}\right] \leq p_0$ *if*

$$\ell > \left\lceil \frac{1}{\log 2} \log \frac{\delta \sum_{i=\mu-\gamma}^{\delta} HG(n_1, N, i, \delta) \sum_{z=\mu-\gamma}^{i} \frac{1}{z}}{p_0} \right\rceil.$$

Proof. We wish to determine the value of ℓ for which $\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] \leq p_0$. From (7.23), it follows that

$$\begin{aligned} \Pr[\mathbf{Z} = z] &= \sum_{x} \frac{\binom{i}{x}\binom{\delta-i}{x-z}}{\binom{\delta}{2x-z}} B(2x - z, \delta, p) && \text{(7.28a)} \\ &< \sum_{x} B(2x - z, \delta, p) && \text{(7.28b)} \\ &< \frac{\delta p}{z} && \text{(7.28c)} \end{aligned}$$

In (7.28b), we used the fact that $\binom{N}{n}\binom{M}{m} < \binom{N+M}{n+m}$. In (7.28c), we used the Chernoff bound to limit the tail sum of the Binomial distribution. Substituting to (7.20) yields,

$$\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] < \sum_{i=\mu-\gamma}^{n_1} HG(n_1, N, i, \delta) \sum_{z=\mu-\gamma}^{i} \frac{\delta p}{z}. \tag{7.29}$$

Limiting the right hand side of (7.29) by p_0 and solving for p results in

$$p < \frac{p_0}{\delta \sum_{i=\mu-\gamma}^{n_1} HG(n_1, N, i, \delta) \sum_{z=\mu-\gamma}^{i} \frac{1}{z}}. \tag{7.30}$$

Substituting $p = 0.5^\ell$ and solving for ℓ completes the proof. □

Corollary 4. *For the open vote model,* $\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] \leq p_0$ *if*

$$\ell \geq \left\lceil \frac{1}{\log 2} \log \frac{n_1}{(\mu - \gamma) p_0} \right\rceil.$$

Proof. The proof follows by using the Chernoff bound to limit the tail probability of the binomial distribution in (3.5). □

From Corollaries 2 and 3, we observe that the required number of symbol votes ℓ drops linearly with the logarithm of p_0. This is also attested by the plots in Figure 7.5, which show the required ℓ as a function of p_0, for various margins μ and number of attacked votes δ (to demonstrate the linear relationship of ℓ with the logarithm of p_0, the ceiling function has not been applied). In Figure 7.5, a total of 20 participants were considered and the voting threshold γ was set to zero (plurality rule). Finally, δ was set to the number of positive votes.

Figure 7.5(a) considers the secret vote model under Strategy 1. As μ increases, fewer symbol votes are necessary to provide the same robustness. However, without knowing the vote intend, the adversary nullifies both "yes" and "no" votes, thus making it harder to close the vote margin. In Figure 7.5(b), we plot ℓ as a function of p_0 for different δ and for $\mu = 4$ under the secret vote model. If few votes are attacked (small δ), the achieved robustness is high for relatively small ℓ. When δ increases, a larger ℓ is needed to achieve the same robustness. However, the adversary's gains diminish beyond a certain δ. As more "yes" votes are initially corrupted, the number of remaining "yes" and "no" votes is balanced, thus becoming equally likely to nullify votes of both types with the increase of δ. Such nullification does not close the voting margin. Figure 7.5(c) considers the open vote model under Strategy 1. Comparing to Figure 7.5(a), we observe that a higher ℓ is necessary to provide the same level of robustness when compared to the secret vote model. This is because the adversary only attacks participants that intend to cast votes in favor of $\mathcal{T}$.

Strategy 2: In the second strategy, the adversary injects energy on both subcarriers assigned to a targeted participant to nullify the participant's vote with certainty. This strategy comes at the expense of increased presence (many subcarriers are attacked). The probability of flipping the voting outcome with Strategy 2 is expressed in Propositions 13 and 14 for the secret and the open vote models, respectively.

Proposition 13. *Under the secret vote model, an internal adversary following Strategy 2 can flip the voting outcome for a decision threshold γ and a margin μ with probability*

$$\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] \quad = \quad \sum_{z=\mu-\gamma}^{\min\{n_1,\delta\}} \sum_{x=\lceil \frac{\delta+z}{2} \rceil}^{\min\{n_1,\delta\}} \frac{\binom{n_1}{x}\binom{n_2}{\delta-x}}{\binom{M}{\delta}}, \tag{7.31}$$

when attempting to nullify δ votes.

Proof. Consider a voting outcome $\mathcal{T}$ with a margin μ with the in favor votes be "yes" votes. Let an internal adversary intend to nullify a total of δ votes. Under the secret vote model, the adversary is unaware of the vote intend of each participant. Therefore, the δ votes are selected at random from the total M votes casted by the participants. Of these M votes, $n_1 = \frac{M+\mu}{2}$ are "yes" votes, whereas the remaining $n_2 = \frac{M-\mu}{2}$ are no votes. The adversary successfully flips the voting outcome if at least μ more "yes" votes are nullified relative to "no" votes, when a total of δ are nullified. Note that under Strategy 2, vote nullification occurs with certainty, because the adversary injects energy on both the subcarriers assigned to a targeted participant. This is independent of the number of symbol votes ℓ. Let $\mathbf{X}$ and $\mathbf{Y}$ be two RVs denoting the number of nullified "yes" and number of nullified "no" votes, respectively. Let also $\mathbf{Z} = \mathbf{X} - \mathbf{Y}$. It follows that

$$\begin{aligned}\Pr[\mathbf{Z} = z] &= \sum_{x-y=z} \Pr[\mathbf{X} = x, \mathbf{Y} = y] \\ &= \sum_{x=\lceil \frac{\delta+z}{2} \rceil}^{\min\{n_1,\delta\}} \frac{\binom{n_1}{x}\binom{n_2}{\delta-x}}{\binom{M}{\delta}}. \qquad (7.32)\end{aligned}$$

In (7.32), we used the hypergeometric pmf to account for the selection of x votes from the n_1 "yes" votes and $x - z$ votes from the n_2 "no" votes, when a total of δ votes are nullified. Note that the adversary only targets the subcarriers assigned to the M participants and ignores any of the unassigned subcarriers if $N > 2M$ (this is not the case for an external adversary). Also, the difference between x and y is fixed to be equal to z, independent of the number of nullified votes δ. From (7.32), we calculate the probability that at least μ more "yes" votes are nullified relative to "no" votes, by summing over all $z \geq \mu$.

$$\begin{aligned}\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] &= \sum_{z \geq \mu-\gamma} \Pr[\mathbf{Z} = z] \\ &= \sum_{z=\mu-\gamma}^{\min\{n_1,\delta\}} \sum_{x=\lceil \frac{\delta+z}{2} \rceil}^{\min\{n_1,\delta\}} \frac{\binom{n_1}{x}\binom{n_2}{\delta-x}}{\binom{M}{\delta}}. \qquad (7.33)\end{aligned}$$

□

Proposition 14. *Under the open vote model, an internal adversary following Strategy 2 can flip the voting outcome for a decision threshold γ and a margin μ with certainty, or $\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] = 1$, when injecting energy in $J \geq \mu - \gamma$ subcarriers.*

Proof. The proof immediately follows by noting that the adversary must nullify $\mu - \gamma$ votes in favor of $\mathcal{T}$ to flip the voting outcome at the tallier. Each of the $\mu - \gamma$ votes is submitted by injecting energy to one of the subcarriers assigned to the corresponding participant. Injecting energy in both those subcarriers nullifies an in favor vote with certainty. Targeting a total of $J = 2(\mu - \gamma)$ subcarriers that correspond to in favor votes (the vote intend is known under an open vote model), nullifies $\mu - \gamma$ in favor votes with certainty, thus flipping the voting outcome. □

Figure 7.5(d) shows the probability of flipping the voting outcome as a function of J for various μ, under the secret vote model. This probability decreases with μ because the adversary must nullify more in favor votes. Moreover, it increases with J. Note that the probability of flipping the voting outcome no longer depends on the security parameter ℓ. This is because the adversary injects energy over both subcarriers assigned to a targeted participants, and therefore nullifies the targeted vote with certainty, irrespective of ℓ.

When Strategy 2 is employed under the open vote model, the voting outcome can be flipped with certainty by attacking a number of votes equal to the vote margin. This is because the energy injection is limited to the subcarriers of participants that intend to cast in favor votes. Nullifying μ of those votes is sufficient to close the voting margin.

Subcarrier sequence preloading: To cope with an internal adversary following Strategy 2, we design a method for concealing the subcarrier assignment between participants. Without knowledge of the subcarriers assigned to others, an internal adversary becomes equivalent to an external one. He can only blindly inject energy on various subcarriers hoping to nullify in favor votes and flip the voting outcome. To hide the subcarriers used by each participant, we modify the setup and vote casting phases as follows.

Setup Phase: In the setup phase, the administrator preloads relevant quantities to the participants and the tallier.

Key generation: The administrator generates keys K_{perm} and $K_{vote,i}$ as described in Section 7.3.1.

Key assignment: The administrator preloads $K_{vote,i}$ to each participant u_i. The administrator preloads $K_{vote,i}$ and K_{perm} to the tallier.

Subcarrier sequence preloading: The administrator computes the subcarrier assignment for each participant u_i by applying pseudo-random function

$$\Pi_F : \{0,1\}^\tau \times [1,N] \times \mathbb{Z}^+ \rightarrow [1..N],$$

to map subcarrier with index p during slot n, to subcarrier $\Pi_F(K_{perm}, p, n)$. For each participant, it computes

$$F_{u_i} = \{(f^0_{u_i}(1), f^1_{u_i}(1)), (f^0_{u_i}(2), f^1_{u_i}(2)), \ldots, (f^0_{u_i}(n), f^1_{u_i}(n)\}$$

where, $f^0_{u_i}(j) = f_{\Pi_F(K_{perm},(2i-1),j)}$, and $f^1_{u_i}(j) = f_{\Pi_F(K_{perm},2i,j)}$. Sequence F_{u_i} is preloaded to participant u_i[1].

Vote Casting Phase: The vote casting phase remains the same as in Section 7.3.3, with the exception of skipping the subcarrier assignment step. By preloading the subcarrier sequence at each participant, an internal adversary u_i cannot infer the subcarrier assignment of any other participant. The adversary is only aware of his own sequence F_{u_i}. Without access to K_{perm}, the adversary can only select the subcarriers where energy is injected at random. In this case, the robustness of PHYVOS under an internal adversary model becomes equivalent to the robustness of PHYVOS under an external adversary, as it is analyzed in Section 7.4.1. Note that a formula adjustment is needed in Proposition 10 to account for the reduction in the number of subcarriers unknown to the adversary. Since u_i is aware of his own subcarrier assignment, it selects to inject energy to J out of the remaining $N-2$ subcarriers (as opposed to J out of N as stated in Proposition 10). Nevertheless, the robustness computation follows along the same lines as in Proposition 10 and therefore, it is omitted.

The subcarrier sequence preloading comes at the expense of extra storage at each participant, which is linear to the number of voting rounds. The storage required to

[1] If preloading is not possible, the sequence F_{u_i} can be generated by the tallier that stores K_{perm}. The tallier can securely communicate F_{u_i} to a participant using $K_{vote,i}$.

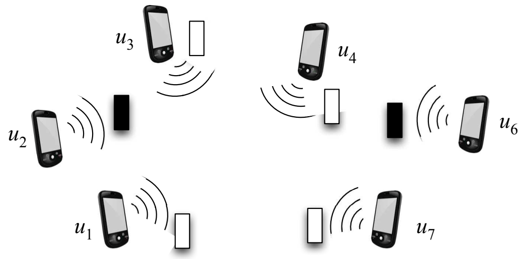


Figure 7.6: The PHYVOS distributed voting scheme. Wireless devices cast their votes to each other using orthogonal subcarriers. Each participant tallies all votes and computes the voting outcome.

support $\mathcal{L}$ voting rounds with ℓ symbol votes per round is equal to $2\lceil\log_2 N\rceil\ell\mathcal{L}$ bits (each voting round consists of ℓ symbol votes casted in one of the two subcarriers indexed by $2\lceil\log_2 N\rceil$ bits). For example, a sequence of 80 Kbytes would support 10^5 voting rounds over 64 subcarriers.

7.5 Voting Without a Centralized Tallier

In this section, we design an implementation of PHYVOS without a centralized tallier. The scenario is depicted in Figure 7.6. A set of six participants co-located within the same collision domain cast their votes. Each participant acts as a tallier by independently tallying the votes casted by other participants and computing the voting outcome. All participants end up with the same voting outcome estimate $\hat{\mathcal{T}}$. To maintain the parallel nature of our PHY-layer voting technique, participants must be capable of simultaneously cast votes and performing the tallying operation. This entails the simultaneous transmission and reception over the OFDM band, that is the operation of each participant in full duplex (FD) mode. We outline two transceiver solutions that enable this concurrent transmission and reception. The first solution exploits self-interference cancellation (SIC) techniques to enable the FD mode. The second solution explores principles similar to OFDMA to allow for the simultaneous vote casting from multiple participants

7.5.1 Full Duplex OFDM

Recent advances on SIC techniques have shown that it is feasible to transmit and receive over the same frequency band [145, 146]. This is achieved by suppressing a significant portion of self interference, using a combination of antenna-based SIC, signal inversion, and RF/digital interference cancellation. In these techniques, the transmitted signal is subtracted from the received signal such that the former does not occupy the dynamic range of the ADC, allowing for the decoding of the incoming signal. For OFDM systems, FD can be realized by independently reducing self-interference at each subcarrier using narrowband cancellation techniques [146, 147].

The operating characteristics of PHYVOS, make the adoption of SIC based FD OFDM easier than its use for the communication of messages. First, each transmitter injects a signal on a single subcarrier, leaving the rest of the subcarriers empty. Thus, the self-interference in other subcarriers is small and primarily limited to the adjacent subcarriers. Applying SIC on the specific subcarrier used to cast a vote further reduces the interference on other subcarriers. Moreover, no signal decoding is necessary. Determination of votes is performed by detecting energy at the output of the FFT block. An imperfect cancellation at subcarrier $f_{u_i}^j$ used by a participant u_i to cast a vote v_i does not affect the tallying of v_i at u_i. Participant u_i is already aware of his own voting intend and does need to decode the symbol transmitted on $f_{u_i}^j$ to determine v_i.

7.5.2 OFDMA

If participants are not equipped with SIC-capable transceivers, FD operation can be achieved by applying OFDMA. Assuming that the transceivers can concurrently operate their transmission and reception radio chains, they can rely on frequency separation to enable the simultaneous vote transmission and reception. Using the adjacent subcarrier method (ASM) [148], participants can form subchannels from adjacent subcarriers so that additional frequency separation is created. In particular, each subchannel consists of three adjacent subcarriers. To cast a vote on a subchannel, energy is injected on the middle subcarrier, using the adjacent subcarriers as guards. Although this approach limits the spectral efficiency of OFDM by essentially converting it to a FDD system, it still provides significant delay reduction for PHY-layer voting relative to message-based voting.

7.5.3 Decentralized PHYVOS

Similar to the centralized tallier scenario, the decentralized PHYVOS consists of four phases: the setup phase, the vote request phase, the vote casting phase, and the tallying phase.

Setup Phase: In the setup phase, the administrator initializes all M participants by preloading K_{perm} to each participant. Note that the pairwise keys $K_{vote,i}$ used for sharing a pairwise secret random sequence between each voter and the tallier are no longer used. The sequences were applied to each symbol vote to conceal the vote-to-subcarrier mapping from internal adversaries (Step 3 of the vote casting phase). When the tallier is replicated at every participant, all sequences R_i must be disclosed to participants, thus negating their security function.

Vote Request Phase: The vote request phase follows the same steps described in Section 7.3.2.

Vote Casting Phase: In the vote casting phase, participants cast and receive votes simultaneously using FD-OFDM. Each vote v_i consists of a series of ℓ symbol votes. The vote casting steps are as follows:

Subcarrier assignment: The subcarrier assignment is performed in the same manner as in Section 7.3.3.

Vote casting: Let voting casting be initiated at slot n_0. To cast a vote $v_i \in \{0, 1\}$, a participant u_i generates ℓ symbol votes $v_i(n_0) = v_i(n_0+1) = \ldots = v_i(n_0+\ell-1) = v_i$. Each $v_i(n)$ is represented by an OFDM symbol with the following values per subcarrier

$$x_k(n) = \begin{cases} \alpha_y, & f_{u_i}^{v_i(n)}(n) \\ 0, & \text{otherwise}, \end{cases} \tag{7.34}$$

where α_y is a randomly selected modulation symbol and $n_0 \leq n < n_0 + \ell$. Note that the placement of energy of either $f_{u_i}^0(n)$ or $f_{u_i}^1(n)$ is solely based on the value of $v_i(n)$.

Vote Tallying Phase: In the vote tallying phase, each participant u_i individually computes the votes of other participants by applying Steps 1-4 outlined in Section 7.3.4. The only difference is in the application of Step 2 for extracting symbol votes. Eq. (7.5) is modified as follows to omit the XORing of the symbol votes with the pseudo-random binary sequence.

$$\hat{v}_i(n) = \begin{cases} 0, & \text{if } p_{u_i}^0(n) > \gamma_D, \quad p_{u_i}^1(n) \leq \gamma_D \\ 1, & \text{if } p_{u_i}^0(n) \leq \gamma_D, \quad p_{u_i}^1(n) > \gamma_D \\ e, & \text{otherwise.} \end{cases} \tag{7.35}$$

7.5.4 Security Analysis

In this section, we briefly sketch the robustness of PHYVOS with a decentralized tallier under an external and internal adversary model.

External adversary: An external adversary is unaware of the cryptographic key K_{perm} used to permute the subcarrier assignment per symbol vote. Therefore, his best strategy is to inject energy on randomly selected subcarriers. Let the adversary inject energy on J subcarriers, as in the case of centralized PHYVOS. Consider the tallying operation occurring at participant u_i. By injecting energy on J subcarriers, the adversary can potentially impact any vote but v_i, because v_i is known to u_i a priori. If v_i is in favor of the voting outcome $\mathcal{T}$, the adversary has to successfully nullify $\mu - \gamma$ votes excluding v_i in order to flip $\hat{\mathcal{T}}$. This probability is given by Proposition 10 by adjusting the number of in-favor votes that can be nullified to

$n_1 = \frac{M+\mu}{2} - 1$. If v_i is against the voting outcome $\mathcal{T}$, the probability of flipping the voting outcome is given by Proposition 10, without adjusting n_1.

Internal adversary: An internal adversary is aware of cryptographic key K_{perm} used by each participant for generating its subcarrier assignment. This allows the adversary to identify the subcarriers used by specific participants to cast votes. Moreover, the subcarrier-to-vote mapping is known because it is no longer randomized by the pairwise secret sequences R_i. The application of these sequences is no longer effective because every participant must be aware of them to correctly tally votes. With full knowledge of the subcarrier assignment, flipping the voting outcome can be achieved by nullifying $\mu - \gamma$ in-favor votes by targeting exactly $\mu - \gamma$ subcarriers.

Although the tally modification cannot be prevented, it is easily detectable by legitimate participants. In-favor participants can determine that their votes are nullified by detecting energy on the opposite subcarrier from the active one. Moreover, the number of nullified votes received by each participant (tallier) is indicative of an ongoing tally modification. In this case, the voting results can be invalidated.

7.6 Voting Overhead

In this section, we compare the voting delay of PHYVOS with the voting delay of message-based voting. Suppose a popular OFDM-based protocol such as 802.11g is used for message-based voting (MV). Each 802.11g packet consists of a 20 μsec preamble (5 OFDM symbols), a 30-byte MAC header and a 4-byte CRC code. Moreover, the vote integrity is protected by a message authentication code based on a secure hash function such as SHA-256 [103]. The message digest size for SHA-256 is 32 bytes. Assuming the highest possible transmission rate for 802.11g, each OFDM symbol can carry 6 bits per subcarrier, times 48 data subcarriers = 36 bytes. Therefore, one vote can be transmitted in 7 OFDM symbols. Ignoring any contention for capturing the wireless medium, participants must wait at least a DCF interframe space (DIFS) between transmitting messages. For 802.11g, DIFS = 13 OFDM symbols. The total delay required to cast M votes becomes

$$D_{MV} = 20M - 13 \ \text{ OFDM symbols.} \tag{7.36}$$

In PHYVOS, up to 26 participants can simultaneously cast their votes using ℓ OFDM symbols (for 52 subcarriers and no pilots). For $M > 26$, a second voting round is required. The value of ℓ is based on the analysis presented in Section 7.4. For our comparison, we set $\ell = 11$ symbols, which yields a robustness level of 10^{-3} (we note that this is an online attack, without any opportunity for repeated trials. Therefore, a robustness of 10^{-3} is acceptable). The total delay required to cast M votes becomes,

$$D_{PHYVOS} = \left\lceil \frac{M}{26} \right\rceil \ell \ \text{ OFDM symbols.} \tag{7.37}$$

Figure 7.7 shows the voting delay as a function of the number of participants M, assuming a typical OFDM symbol duration of 4μsec. PHYVOS reduces delay by one order of magnitude for $M = 11$ and two orders of magnitude for $M = 50$. Note that for $M = 26$, the MV incurs a delay of at least 2 sec.

We note that in most modern OFDM systems the number of available subcarriers could be substantially higher than 52. For instance, the number of subcarriers in LTE exceeds 300 and can reach up to 1,200 when the allocated bandwidth is 20 MHz. Therefore, a much larger number of participants can be simultaneously supported, although we do not anticipate that this number will be large for one-hop scenarios. In the event that multiple rounds are needed to accommodate the number of voting participants, the individual delay until a each participants casts its vote does not affect the voting delay, which is defined as the delay until all votes are casted. If an application requires rectifying the unfairness in the individual voting delay, a round robin approach can be used to alternate between voting groups on every voting round.

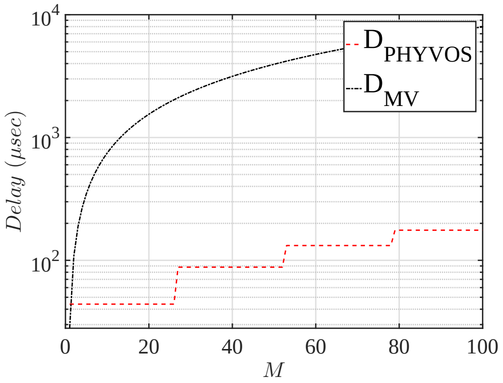


Figure 7.7: Voting overhead as a function of M for message-based voting (MV) and PHYVOS.

7.7 Practical Considerations and Implementation

7.7.1 Frequency Synchronization

Radio oscillators do not operate at the same nominal frequency due to manufacturing imperfections. This frequency misalignment is known as carrier frequency offset (CFO). OFDM systems are particularly sensitive to CFO due to the subcarrier orthogonality requirement. The CFO has two critical effects on demodulation. First, subcarriers are no longer orthogonal causing inter-carrier interference (ICI) and reducing the SNR. Second, symbols at each subcarrier appear arbitrary rotated in the constellation. Finally, a large CFO can cause a subcarrier shift at the receiver, whereby a symbol transmitted over subcarrier f_i is mapped to f_j. This shift occurs if the CFO is larger than the subcarrier spacing [149, 150].

To mitigate the impact of CFO in practical systems, receivers estimate the CFO using the preamble transmitted with every packet. In PHYVOS, no preamble is present with the transmission of votes to save on messaging overhead. However, the lack of frequency synchronization does not impact the correct vote estimation, because *no demodulation is performed.* Any symbol rotation in the constellation map does not affect the energy estimation on a given subcarrier. After all, the

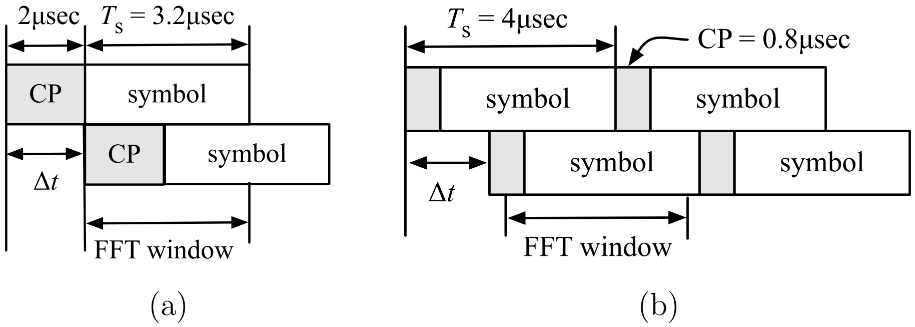


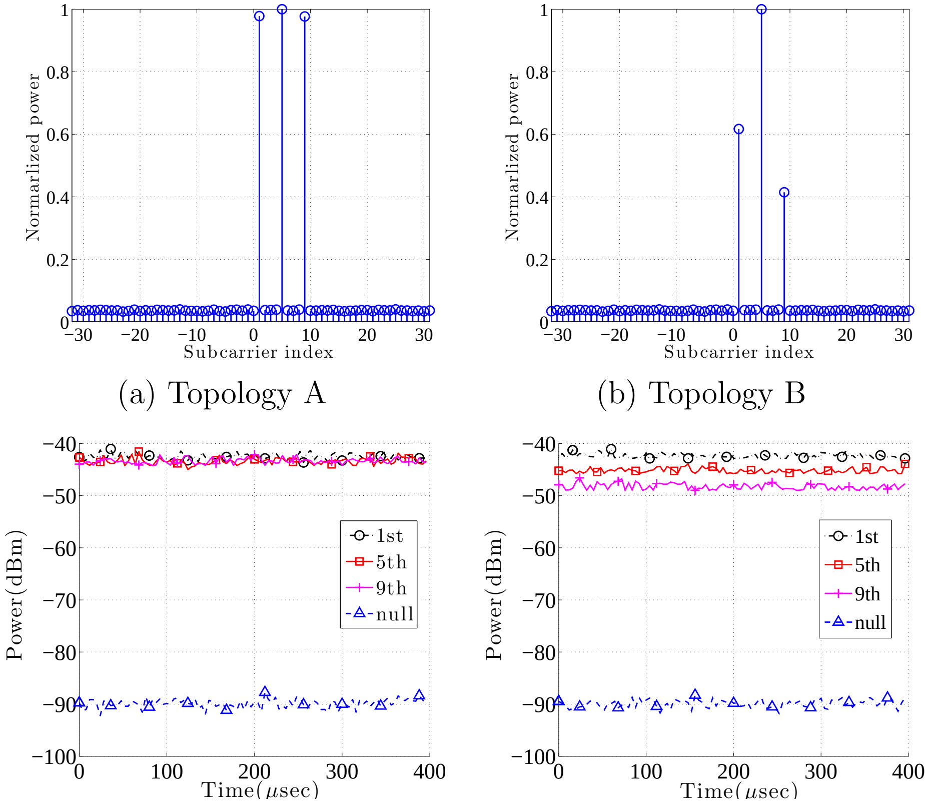
Figure 7.8: (a) Increasing the CP, (b) casting a symbol vote in two symbol durations.

symbol transmitted to realize a vote is selected at random and does not convey any information. Furthermore, for a CFO that does not cause a subcarrier bin shift, the strongest ICI component comes from adjacent subcarriers. To limit ICI, the subcarriers assigned to each participant can be spaced as far as the number of participants allows. For instance, for 10 voters and 64 subcarriers, every 3rd subcarrier is used to cast a vote.

7.7.2 Time Synchronization

Another practical problem for PHYVOS is that symbol votes do not reach the tallier perfectly synchronized. Differences in propagation delay and device clock drifts can cause a time misalignment between the symbol votes casted by each device. This misalignment will affect the set of samples that fall within the FFT window of the Fourier transform applied at the receiver for extracting the spectral components of the OFDM signal. This is similar to *symbol bleeding* caused in OFDM systems when delayed copies of OFDM symbols arrive at the receiver due to multipath effects. The solution applied in OFDM is to append a cyclic prefix (CP) to every symbol, which is in the order of 0.8 μsec.

For PHYVOS, the time misalignment Δt between symbols at the receiver can be greater than 0.8 μsec. For a typical WiFi range of 300m, the propagation delay difference between two devices can by up to 1μsec. Moreover, the typical clock error



(a) Topology A (b) Topology B

(c) Topology A (d) Topology B

Figure 7.9: (a), (b) Normalized average received power per subcarrier, (c), (d) received power per subcarrier as a function of time.

for modern clocks is well below 5ppm [151]. If clock synchronization is performed every 100msec (typical beacon transmission period for WiFi base stations), the expected clock error between two devices can be up to 1μsec, making the total time misalignment $\Delta t \leq 2\mu$sec.

To cope with the symbol misalignment, we can extend the CP duration to 2μsec to account for the maximum expected Δt. The increase in CP comes at the expense of a higher overhead to cast a symbol vote (5.2 μsec vs. 4 μsec). Note that the increased CP duration is adopted only for vote casting and is not part of the normal OFDM operation for data transmissions. Alternatively, to maintain compatibility with the current OFDM specifications, we can extend the symbol vote duration to two OFDM symbols, without increasing the CP duration. This solution comes at the expense of doubling the overhead for casting a symbol vote. A similar solution was adopted in [138]. The two solutions are shown in Figure 7.8.

7.7.3 PHYVOS Implementation

Testbed setup: We implemented PHYVOS on NI USRPs 2921 devices, operating in the 2.4 GHz band over a 39.6 MHz spectrum. A total of four radios were at our disposal. Under normal operation, three radios operated as voters, whereas one radio operated as the tallier. One radio was switched to an attacker role for adversarial scenarios. Voter radios were placed in a LoS configuration at varying distances from the tallier within an office environment. We divided the 39.6 MHz spectrum to 64 subcarriers. To cast a symbol vote, each radio used BPSK modulation to transmit a random symbol at the designated subcarrier. The CP value was set to 0.8 μsec, as the time synchronization error between the different radios was relatively small. We used a 64-point FFT to collect the symbol votes from each subcarrier. The transmission power of each radio was set to 20 dBm (0.1 W).

Selection of threshold γ_D: In the first experiment, we investigated the selection of the power threshold γ_D used in eq. (7.5) for detecting votes. We assigned the 1st, 5th, and 9th subcarrier to each of the three voter radios. Each voter casted 1,000 symbol votes at its designated subcarrier by transmitting 1,000 BPSK symbols. The rest of the subcarriers remained null. A time gap of100 msec was imposed between two consecutive votes. Figure 7.9(a) shows the normalized magnitude of the FFT output at the tallier, averaged over the 1,000 transmitted symbols when the three voters are placed 5ft away from the tallier (topology A). Figure 7.9(b) shows the same results when the three voters are at 5ft, 10ft, and 15ft away from the tallier (topology B).

Figure 7.9(c) and 7.9(d) show the received power as a function of time for 100 consecutive symbols. For topology A, the power of active subcarriers is approximately -42dBm, whereas the power of null subcarriers is -90dBm. The recorded -90dBm value for the null subcarriers is well above the noise floor due to the operation of nearby devices over the ISM band. For topology B, the received power from the farthest radio dropped to -49dBm. Based on the recorded values, we set the threshold γ_D for the detection of a symbol vote to -80dBm, which is well above the receiver sensitivity.

Time synchronization: In the second experiment, we studied the effect of time synchronization on the correct operation of PHYVOS. The experimental setup is shown in Figure 7.10(a). We used one USRP as the FC, while three USRPs were setup as voting participants. We set the CP value to 2.0μsec, the FFT window to 1.2μsec, and varied the time synchronization error between the participants. This was achieved by adjusting the firing times of the USRP devices for symbol transmissions, while the USRPs were placed at different distances from the FC. The three participants u_1, u_2, u_3 were placed as follows: u_1 was placed at 15ft from the FC with a LoS channel, u_2 was placed at 10ft from the FC with a LoS channel, whereas u_3 was placed at 5ft from the FC, but with an obstruction on the LoS path. This created different profiles of synchronization offset for different users due to multipath and also clock errors. For each synchronization offset (Δt), we transmitted 10^6 votes.

Further, we performed the experiment for two subcarrier allocations. In the first allocation, the USRP devices were assigned non-adjacent subcarriers, (1,2), (9,10), and (15,16) for submitting yes/no votes. In the second allocation, USRPs were assigned adjacent subcarriers (1,2), (3,4), and (5,6). In Figure 7.10(b), we show the fraction of erroneously received votes as a function of the maximum synchronization error Δt between any two devices. We note that as long as the CP duration is larger than Δt, votes are correctly inferred despite the symbol misalignment. The scenario with non-adjacent subcarriers achieves slightly better performance, as the sample misalignment does not impact adjacent votes.

We repeated the above experiment for the topology of Figure 7.10(c), where u_3

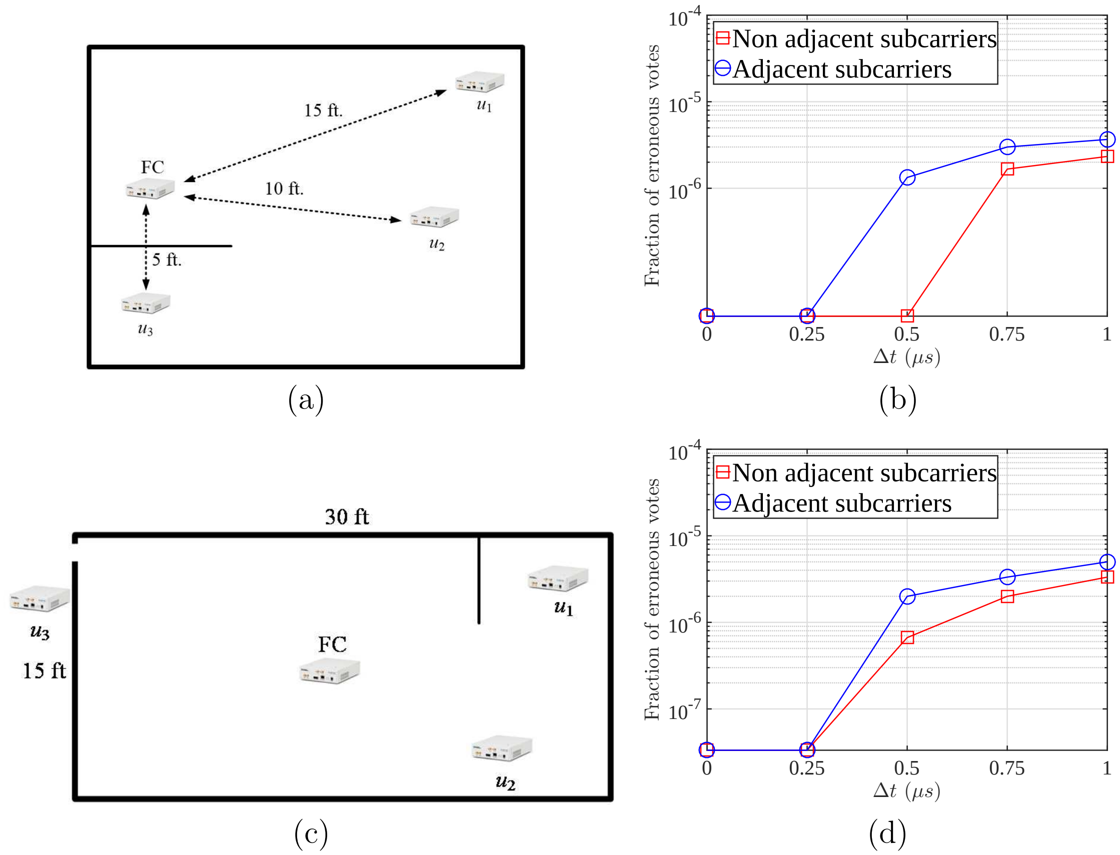
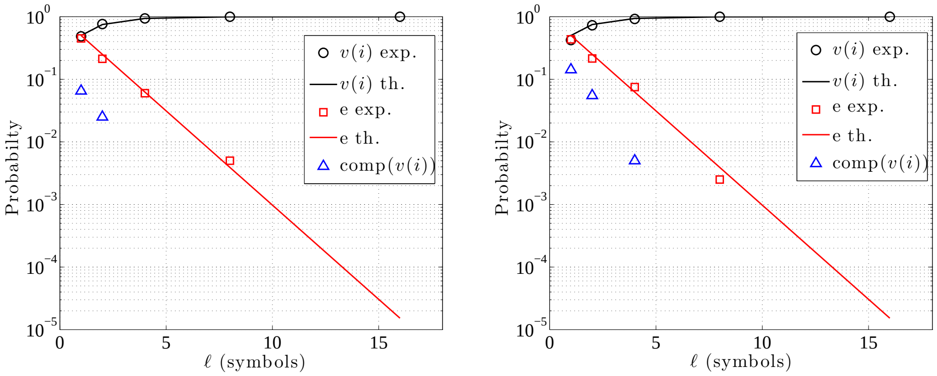


Figure 7.10: (a) The USRP topology used to evaluate the effect of time synchronization, (b) fraction of erroneously decoded votes at the receiver as a function of the synchronization error (Δt) between participants, c) the USRP topology used to evaluate the effect of NLoS paths, and (d) fraction of erroneously received votes as a function of the synchronization error (Δt) for the topology of Fig 7.10(c).

was placed on the outside of the room that housed the FC, thus obstructing the LoS path. In Fig 7.10(d), we show the fraction of lost votes as a function of Δt. We observed similar results to the performance under the topology of Figure 7.10(a), indicating that the use of a longer CP alleviates the misynchronization phenomenon even for NLoS channels.

Voting in the presence of an internal adversary: In the third experiment, we implemented Strategy 1 for an internal adversarial. One of the three USRPs was assigned the role of an internal attacker that is aware of the subcarrier assignment to



(a) (b)

Figure 7.11: (a) Probability of tallying the correct vote $v(i)$, having an inconclusive vote e, or flipping the vote to comp($v(i)$), and (b) probability of tallying the correct vote $v(i)$, having an inconclusive vote e, or flipping the vote to comp($v(i)$).

other voters. Voter #1 was assigned the 1st and 2nd subcarrier while voter #2 was assigned the 5th and 6th subcarrier. For each symbol vote, the attacker randomly selected one subcarrier per voter and injected a random symbol in order to nullify or flip the casted vote (Strategy 1). The experiment lasted for 10^6 symbol votes. Figure 7.11(a) shows the probability of tallying the correct vote $v(i)$, having an inconclusive vote e, or flipping the vote to comp($v(i)$), as a function of the security parameter ℓ for topologies A and B. The theoretical values for tallying the correct vote $v(i)$, and having an inconclusive vote e are also shown (solid lines). The theoretical values are computed according to equation (7.19).

We observe that the experimental values are in close agreement with the theoretical ones. As expected, the probability of tallying the correct vote rapidly converges to one with the increase of ℓ, whereas the probability of an inconclusive vote becomes small (zero for $\ell > 8$). In our experiments, some votes were actually flipped indicating a drop in the received power on a designated subcarrier to a value smaller than γ_D for ℓ consecutive symbol votes. However, this occurred with very low probability and was not observed at all when $\ell > 2$. The results were similar for topology B (see Figure 7.11(b)), with a slight increase in the probability of flipping a vote. This was primarily observed due to the near-far effect for the most distant voter (placed at 15ft from the tallier).

7.7.4 Simulated Experiments

The USRP experiments involved a small number of devices and were primarily used to study the implementation nuances of simultaneous vote casting. In this section, we perform simulated voting experiments with a large number of participants.

Simulation setup: We simulated PHYVOS using MATLAB R2015B [152]. We initially considered 26 participants casting votes over 52 subcarriers to a FC. We repeated some experiments for 100 participants. The wireless channel between the tallier and each participant was simulated by a Rician fading model with maximum path delay 1.5×10^{-6} sec, a *K-factor* equal to two, and a LoS SNR equal to 15dB. The Rician channel was selected because it is representative in many one-hop topologies. To cast a symbol vote, participants randomly selected a QPSK symbol. The symbol vote detection threshold γ_D was set to -80 dBm. A plurality vote criterion ($\gamma = 0$) was applied to compute the voting outcome.

Vote nullification due to channel imperfections: In the first set of experiments, we measured the probability of unintentional vote nullification due to wireless channel imperfections. In the absence of an adversary, we varied the SNR of the participant-tallier channel and measured the number of nullified votes at the tallier. Each vote consisted of three symbol votes. Figure 7.12(a) shows the CDF of the nullified votes for different SNRs. We observe that even at low SNR values (≤ 10 dB), less than four out of the 26 votes are nullified due to fading, with probability over 95%.

We also measured the number of unintentionally nullified votes due to CFO and time offsets. These effects are discussed in Section 7.7.1 and 7.7.2. Each participant was randomly assigned a CFO of either 0 KHz or CFO_{max}. We opted to combine participants with and without CFO to allow the maximum frequency misalignment between certain subcarriers at the tallier. Figure 7.12(b) shows the CDF of the nullified votes for various $\text{CFO}_{\max}$. For typical CFO values, less than two out of 26 votes are nullified in 95% of the observed runs. Vote nullification occurs when a sufficient amount of energy is leaked to adjacent subcarriers due to the CFO. The effect of the CFO can be mitigated if the tallier compensates for it before vote tallying. The tallier can record the CFO of each participant using the preambles of prior packet transmissions. Note that CFO estimation must be repeated infrequently, as it varies very slowly with time.

Furthermore, we simulated the impact of time synchronization errors caused by the misalignment of symbol votes due to time offsets (Δt). Each participant was assigned a time offset of either 0 μsec or $\Delta t_{\max}$ μsec at random. The tallier used the two symbol vote estimation technique outlined in Section 7.7.2 to compensate for the symbol time misalignment. Figure 7.12(c) shows the CDF of the nullified votes for varying $\Delta t_{\max}$. We observe that two symbols for the symbol vote estimation eliminates the impact of misalignment.

In Figure 7.12(d), we show the CDF of the nullified votes when fading, CFO, and time misalignment are all present in the same experiment. We observe that under typical values (SNR = 15dB, CFO = 25kHz and $\Delta t = 1$ μsec), less than one votes are nullified, on average, with probability over 95%. In worse conditions (SNR = 5dB, CFO = 100kHz and $\Delta t = 1.5$ μsec), less than six votes are nullified with probability over 95%. This CDF shift is primarily due to the low SNR. We use Figure 7.12(d) to set the threshold γ_{null} to six votes. Recall that γ_{null} is used to detect the presence of an adversary if an unusual number of votes are nullified at the tallier.

Finally, we performed a simulated experiment to evaluate the effects of various channel models on vote correctness. We measured the number of erroneously received votes at the FC for an AWGN channel, a Rayleigh channel with a maximum path delay of $1.5\mu sec$, a Rician channel with a K factor of 2, and a maximum path delay of $1.5\mu sec$ and a Nakagami-m channel with fading factors 0.5 and 10. A total

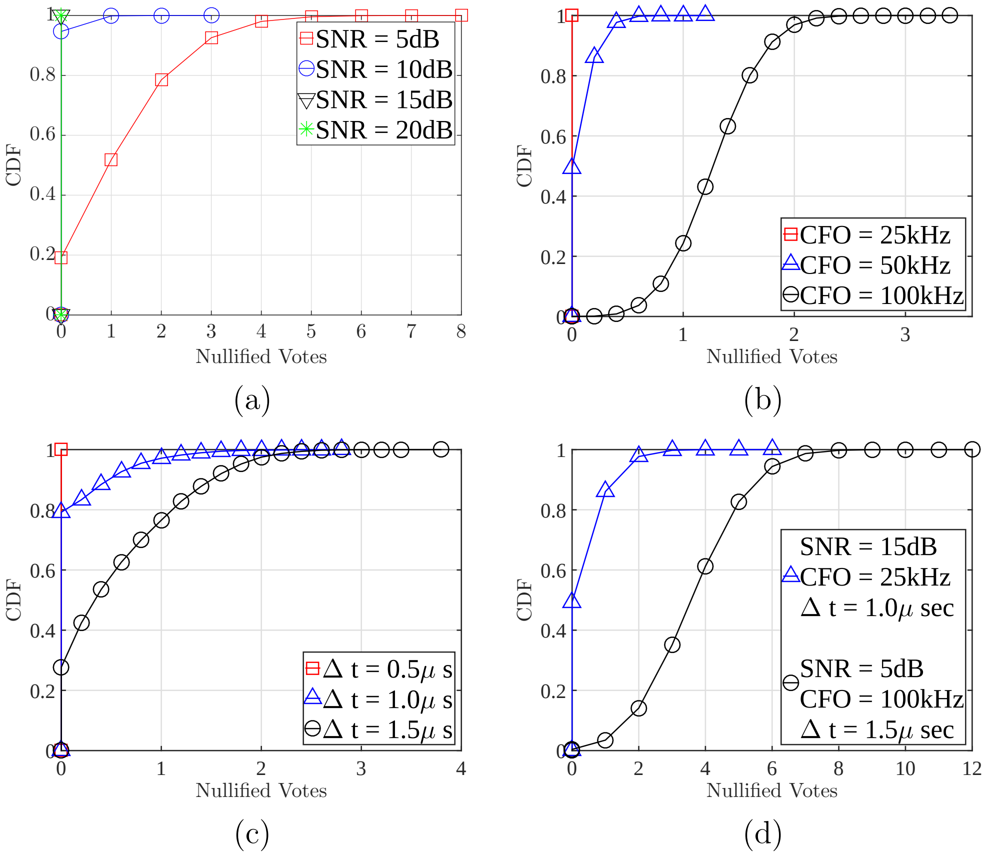


Figure 7.12: (a) CDF plot of number of nullified votes due to wireless channel noise for varying channel SNR, (b) CDF plot of number of nullified votes received due to carrier frequency offset error for varying CFO, (c) CDF plot of number of nullified votes received due to synchronization error for varying time offset, and (d) CDF of the nullified votes when the fading, CFO, and time misalignment phenomena are all combined in the same experiment.

of 10^6 votes per participant were transmitted. In Figure 7.13, we show the fraction of erroneous votes received at the FC as a function of the transmit power in dBm. It can be observed that the vote error remains below 10^{-5} for all transmit powers and it drops with the power increase. The Nakagami-m channel with a fading factor of 0.5 yields the worst performance, but the error is still quite low and does not significantly affect the energy-based vote detection.

Effect of varying participant distances: In this scenario, we placed two par-

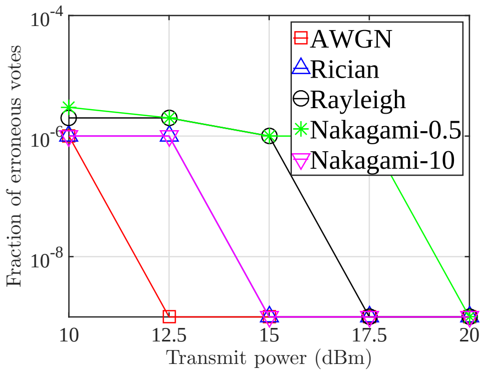


Figure 7.13: Fraction of incorrectly received votes as a function of the transmission power for various wireless channel models.

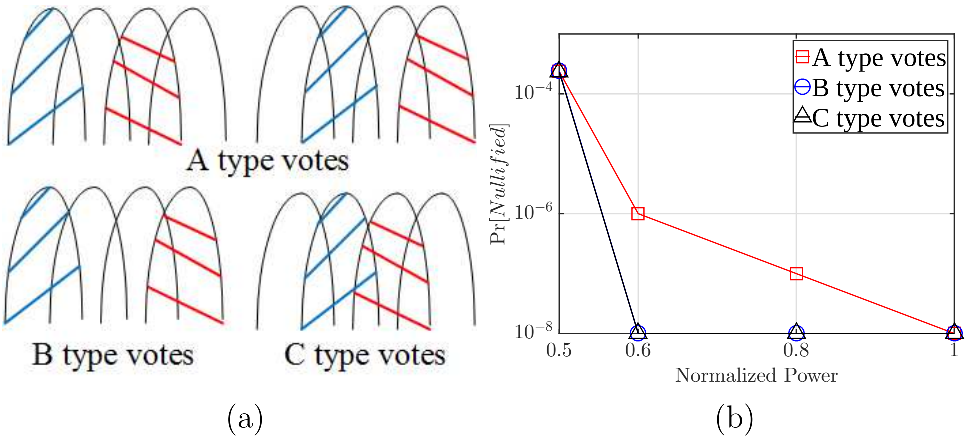
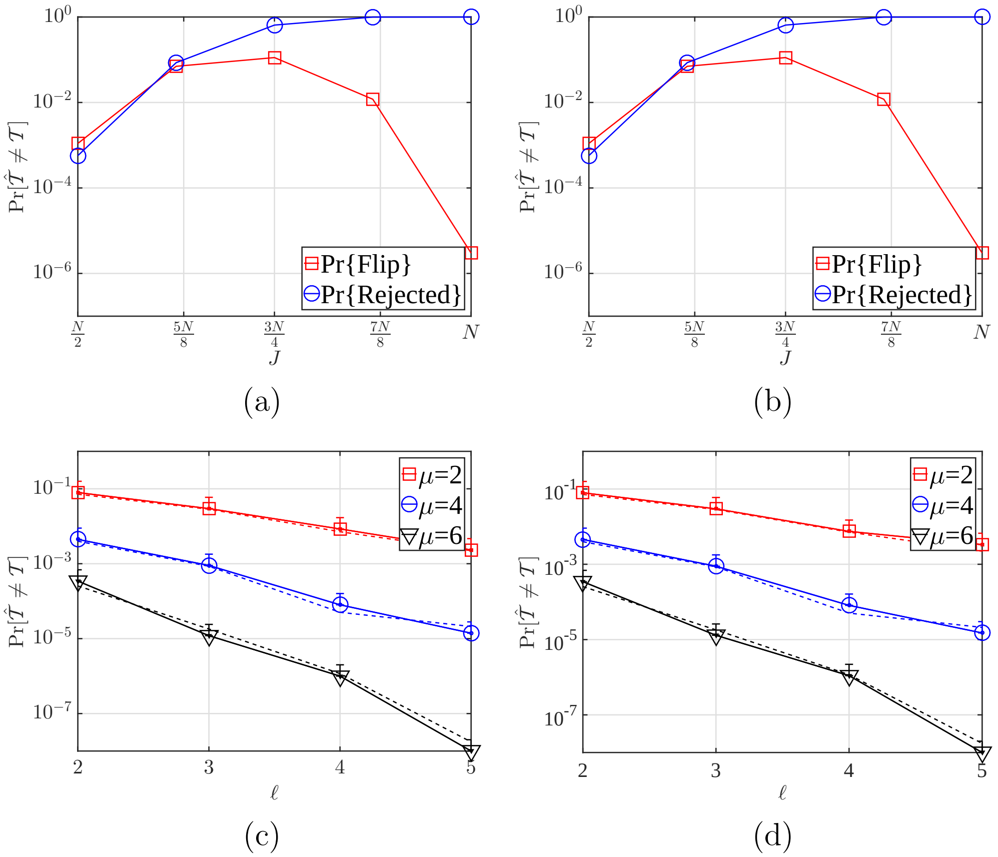


Figure 7.14: (a) Energy assignment to subcarriers for all voting combinations, and (b) probability of votes received incorrectly plotted against normalized power of votes received for all possible voting combination.

ticipants at different distances from the tallier in order to vary the received power ratio between subcarriers at the tallier. We considered the subcarrier assignments shown in Figure 7.14(a). Votes of type A represent cases where two participants inject energy on subcarriers separated by a single subcarrier, votes of type B represent cases where the subcarrier separation is equal to two, whereas votes of type C represent cases where the two participants inject energy on adjacent subcarriers. Figure 7.14(b) shows the probability of vote nullification for any of the two participants as a function of the power of the more distant participant, normalized over the power of the closest participant. The probability of vote nullification remains low even when the power of the distant participant is half of the power of the closest one. Votes of type A have the highest probability of being nullified, because energy from two active subcarriers "bleeds" into a common adjacent empty subcarrier. On the other hand, votes of type B and C exhibit the same probability of vote nullification, because only one active subcarrier "bleeds" into an inactive one. As the transmission powers between the participants become equal, the probability of vote flipping attains very low values.

External adversary: In the third set of experiments, we evaluated the robustness of PHYVOS against an external adversary. The adversary attempted to flip the voting outcome by injected energy to J randomly selected subcarriers. The tallier used the threshold γ_{null} to detect an ongoing attack, if a large number of votes are nullified. We also fixed the number of symbol votes to $\ell = 3$ and the voting margin to $\mu = 3$. Figure 7.15(a) shows the tradeoff between probability of flipping $\mathcal{T}$ and rejecting the voting round as a function of J. As J increases, the probability of flipping the voting outcome improves for the adversary until J equals 3/4 of the available subcarriers. Any further increase of J has a negative effect. This is because the adversary nullifies votes that oppose the voting outcome. On the other hand, the probability of rejecting the voting round strictly increases with J. For the value of J that maximizes the probability of flipping the voting outcome ($\Pr[\hat{\mathcal{T}} \neq \mathcal{T}] = 0.1$, for $J = \frac{3N}{4}$), the voting round is rejected with probability 94.2%.

To verify that PHYVOS is scalable, we repeated the simulated experiments for 100 participants who casted votes over 52 subcarriers. As there are only 52 subcarriers available, the participants were divided to three groups of size 26 and one group of size 22. Participants of the same group casted their votes simultaneously using ℓ symbol votes, requiring a total of 4ℓ symbols to complete a voting round. Figure 7.15(b) shows the tradeoff between probability of flipping $\mathcal{T}$ and rejecting the voting round as a function of J. We observe that the increased number of



(a)

(b)

(c)

(d)

Figure 7.15: (a) Probability of flipping voting outcome and rejecting a voting round as a function of the number of attacked subcarriers (J) in presence of an external adversary, and for 26 participants, (b) probability of flipping voting outcome and rejecting a voting round as a function of the number of attacked subcarriers (J) in presence of an external adversary, and for 100 participants, (c) probability of flipping the voting outcome as a function of the number of symbol votes (ℓ) in presence of an internal adversary, and for 26 participants, and (d) probability of flipping the voting outcome as a function of the number of symbol votes (ℓ) in presence of an internal adversary, and for 100 participants.

participants does not qualitatively affect the robustness of PHYVOS.

Internal adversary: In the fourth set of experiments, we evaluated the robustness of PHYVOS against an internal adversary when applying Strategy 1. Using his knowledge of the subcarrier assignment, the adversary injected energy at one of the two subcarriers assigned per participant. In Figure 7.15(c), we show the probability of flipping the voting outcome as a function of number of symbol votes ℓ, and for $\mu = \{2, 4, 6\}$. Solid lines indicate the values obtained via simulation, whereas dotted lines show the theoretical values calculated using (7.20). For the simulation results, we also plot the upper bound of the 95% confidence intervals (the lower bounds are omitted due to the log scale on the Y axis). The simulation results verify the theoretical analysis for the probability of flipping the voting outcome. Using larger values of ℓ allows the tallier to substantially reduce this probability. We repeated our experiments for 100 participants who casted their votes in groups. Figure 7.15(d) shows similar results to Figure 7.15(c). This is expected, as only 26 participants vote at every slot.

7.8 Related Work

The use of voting for improving reliability has been studied since the 1950s [153], with a long literature on various reliability and efficiency aspects (e.g., [154–159]). Levitin proposed a weighted mechanism for binary voting where each vote is weighted based on the participant's identity [155]. The author showed that for participants with different decision times, a tradeoff exists between reliability and delay. He proposed an algorithm to maximize reliability under a time constraint. Barbara and Molina studied the reliability of voting mechanisms, when participants are divided into groups and are assigned a number of votes [157]. The group with the voting majority is prioritized to perform critical system operations. They proposed several vote assignment heuristics to improve the overall system reliability. Kwiat *et al.* examined three binary voting rules for fault tolerance and evaluated the resulting reliability and security [158]. They proposed a random selection algorithm for computing the voting outcome from a set of votes that contain malicious ones. We emphasize that PHYVOS implements a PHY layer vote casting mechanism that guarantees vote integrity. The voting rules (majority, random selection, number of votes per participant, vote weights, etc.), which is the subject of most previous studies in reliability and fault-tolerance, is complementary to our method.

In the context of wireless networks, voting finds wide application to data fusion, intrusion detection and secure localization in WSNs [45, 160–162], real-time coordination in multi-agent systems [47], and fault-tolerant protocols [163, 164] The de facto voting mechanism adopted in these works is message-based voting, in which votes are casted through messaging. Message-based voting also facilities the integration of security measures for preventing the manipulation of the voting outcome. Voters can be authenticated, and vote integrity can be verified using standard cryptographic primitives such as digital signatures, message authentication codes, and digital certificates [103]. Compared to message-based voting, PHYVOS requires significantly less communication overhead, without sacrificing robustness to vote manipulation.

From an implementation standpoint, the most relevant works to ours are presented in [137, 138]. In [138], Dutta *et al.* proposed SMACK, an acknowledgment scheme for implementing a reliable broadcast service. Similar to PHYVOS, SMACK exploits the subcarrier orthogonality of OFDM to allow the simultaneous submission of acknowledgements in response to a broadcast message transmitted by a single source. In [137], Rahul *et al.* proposed SourceSync, a distributed wireless architecture that explores sender diversity in OFDM. SourceSync enables the reception and *demodulation* of OFDM symbols composed of symbol transmissions over individual subcarriers by a diverse set of senders. Contrary to SMACK and PHYVOS, SourceSync can demodulate the combined OFDM symbol and retrieve the individual data streams of each sender. This capability comes at the expense of complex symbol-level synchronization and channel estimation at the senders, performed through the transmission of preambles.

Recently, the infeasibility of erasing energy from a wireless channel was challenged. Pöpper *et al.* showed that under stable and predictable channel conditions (e.g., LOS), an attacker utilizing a pair of directional antennas for relaying the inverse of the received signal could cancel a signal at a targeted receiver [14]. Such powerful signal cancellation attacks are hard to launch in practice against PHYVOS due to the multiple wireless channels used by the participants for the simultaneous communication with the tallier. Moreover, channel estimation of any of those channel within the channel coherence time becomes difficult without the transmission of preambles.

7.8.1 Chapter Summary

We presented PHYVOS, a secure and fast PHY-layer voting scheme for wireless networks. In PHYVOS, no explicit messaging is necessary. Participants cast their votes simultaneously by exploiting the subcarrier orthogonality in OFDM. PHYVOS is aimed at reducing the delay overhead for wireless applications where secure voting is time-critical. We analyzed the robustness of PHYVOS against both external and internal adversaries who aim at altering the voting outcome at the tallier. We showed that PHYVOS maintains the integrity of the voting outcome with high probability, without using cryptographic primitives. We extended PHYVOS to a decentralized operation scenario, in which participants can determine the voting outcome without the presence of a centralized tallier. We implemented PHYVOS on the USRP platform and verified the robustness properties via experimentation and simulations.

CHAPTER 8

CONCLUSIONS

In this dissertation, we developed techniques for bootstrapping trust between devices which do not have pre-shared secrets. In particular, we focused on in-band trust establishing techniques that are resistant to both active (MitM) and passive adversaries. To achieve the robustness we proposed secret-free authentication and message integrity techniques that rely on a combination of applied cryptography and PHY-layer properties. Our main achievements and finding are summarized as follows.

In Chapter 3, we proposed a new PHY-layer integrity protection scheme called HELP that is resistant to signal cancellation attacks. Our scheme operates with the assistance of a helper device that has an authenticated channel to the A. The helper is placed in close proximity to the legitimate device and simultaneously transmits at random times to allow the detection of cancellation attacks at the A. We showed that a pairing protocol such as the DH key agreement protocol using HELP as integrity protection primitive can resist MitM attacks without requiring an authenticated channel between D and the A. This was not previously feasible by any of the pairing methods if signal cancellation is possible. We studied various implementation details of HELP and analyzed its security. Our protocol is aimed at alleviating the device pairing problem for IoT devices that may not have the appropriate interfaces for entering or pre-loading cryptographic primitives.

In Chapter 4, we propose SFIRE a secret-free protocol that achieves the secure pairing of COTS wireless devices with a hub. Compared to the state-of-the-art, SFIRE does not require any out-of-band channels, special hardware, or firmware modification, thus it is applicable to any COTS device. We showed that SFIRE is resistant to the most advanced active signal manipulations that include recently demonstrated signal nullification at an intended receiver. These security properties are achieved in-band with the assistance of a helper device and by using the RSS fluctuation patterns to build a robust RSS authenticator. We performed extensive theoretical analysis and attested the finding with experiments using COTS devices and USRP radios and validated the security and performance of the proposed protocol.

We presented VERSE in Chapter 5, a new PHY-layer group message integrity verification primitive resistant to MitM attacks over the wireless channel. We exploit the existence of multiple devices that act as verifiers of the protocol transcript for integrity protection. When three or more devices perform an integrity check, it is infeasible for the adversary to simultaneously manipulate the wireless signal at all devices, based on geometrical constraints. We presented a DH-based device bootstrapping protocol that utilized VERSE, which only requires in-band communications with minimal human effort during initialization. We formally prove the security of both VERSE and the bootstrapping protocol against active attacks. With a real-world USRP testbed, we experimentally validated our theoretical results by showing that an increasing number of devices significantly weakens the adversary's ability to successfully manipulate wireless signals. This is in contrast to prior state-of-the-art where the attacker's success probability increases with the number of devices.

In Chapter 6, we addressed the problem of the verifying the integrity of ADS-B navigation information without modifying the ADS-B standard. We proposed a PHY-layer verification method that exploits the Doppler spread phenomenon and the short coherence time of the channel between a prover aircraft and verifier aircraft to verify the velocity claims of the prover. The solution proposed in this work can be applied independently of the ADS-B standard. We further related the velocity claims to location claims through simple kinematic equations. We analyzed the security of our verification scheme and showed that it is equivalent to solving underdefined quadratic equation systems which are known to be hard.

We presented PHYVOS in Chapter 7, a secure and fast PHY-layer voting scheme for wireless networks. In PHYVOS, no explicit messaging is necessary. Participants cast their votes simultaneously by exploiting the subcarrier orthogonality in OFDM. PHYVOS is aimed at reducing the delay overhead for wireless applications where secure voting is time-critical. We analyzed the robustness of PHYVOS against both external and internal adversaries who aim at altering the voting outcome at the tallier. We showed that PHYVOS maintains the integrity of the voting outcome with high probability, without using cryptographic primitives. We extended PHYVOS to a decentralized operation scenario, in which participants can determine the voting outcome without the presence of a centralized tallier. We implemented PHYVOS on the USRP platform and verified the robustness properties via experimentation and simulations.

REFERENCES

[1] Ming Li, Wenjing Lou, and Kui Ren. Data security and privacy in wireless body area networks. *IEEE Wireless communications*, 17(1), 2010.

[2] Yin Minn Pa Pa, Shogo Suzuki, Katsunari Yoshioka, Tsutomu Matsumoto, Takahiro Kasama, and Christian Rossow. Iotpot: A novel honeypot for revealing current iot threats. *Journal of Information Processing*, 24(3):522–533, 2016.

[3] Vivian Genaro Motti and Kelly Caine. Users' privacy concerns about wearables. In *Proc. of International Conference on Financial Cryptography and Data Security*, pages 231–244. Springer, 2015.

[4] Ke Wan Ching and Manmeet Mahinderjit Singh. Wearable technology devices security and privacy vulnerability analysis. *Int. J. Netw. Secur. Appl*, 8(3):19–30, 2016.

[5] Chantal Lidynia, Philipp Brauner, and Martina Ziefle. A step in the right direction–understanding privacy concerns and perceived sensitivity of fitness trackers. In *Proc. of International Conference on Applied Human Factors and Ergonomics*, pages 42–53. Springer, 2017.

[6] Alexander Wieneke, Christiane Lehrer, Raphael Zeder, and Reinhard Jung. Privacy-related decision-making in the context of wearable use. In *Proc. of PACIS*, page 67, 2016.

[7] Christian Biener, Martin Eling, and Jan Hendrik Wirfs. Insurability of cyber risk: an empirical analysis. *The Geneva Papers on Risk and Insurance-Issues and Practice*, 40(1):131–158, 2015.

[8] The Guardian. DDoS attack that disrupted internet was largest of its kind in history, experts say, 2016.

[9] M-Elisabeth Paté-Cornell, Marshall Kuypers, Matthew Smith, and Philip Keller. Cyber risk management for critical infrastructure: a risk analysis model and three case studies. *Risk Analysis*, 38(2):226–241, 2018.

[10] Maria Tcherni, Andrew Davies, Giza Lopes, and Alan Lizotte. The dark figure of online property crime: Is cyberspace hiding a crime wave? *Justice Quarterly*, 33(5):890–911, 2016.

[11] Krishna Sampigethaya, Radha Poovendran, Sudhakar Shetty, Terry Davis, and Chuck Royalty. Future e-enabled aircraft communications and security: The next 20 years and beyond. *Proc. of the IEEE*, 99(11):2040–2055, 2011.

[12] Krishna Sampigethaya and Radha Poovendran. Security and privacy of future aircraft wireless communications with offboard systems. In *Proc. of the Communication Systems and Networks Conference*, pages 1–6, 2011.

[13] Andrei Costin and Aurélien Francillon. Ghost in the air (traffic): On insecurity of ADS-B protocol and practical attacks on ADS-B devices. *Black Hat USA*, 2012.

[14] Christina Pöpper, Nils Ole Tippenhauer, Boris Danev, and Srdjan Capkun. Investigation of signal and message manipulations on the wireless channel. In *Proc. of European Symposium on Research in Computer Security*, pages 40–59. Springer, 2011.

[15] H-A Wen, T-F Lee, and Tzonelih Hwang. Provably secure three-party password-based authenticated key exchange protocol using weil pairing. *IEEE Proceedings-Communications*, 152(2):138–143, 2005.

[16] Donald McCallie, Jonathan Butts, and Robert Mills. Security analysis of the ADS-B implementation in the next generation air transportation system. *International Journal of Critical Infrastructure Protection*, 4(2):78–87, 2011.

[17] Duncan De Borde. Two-factor authentication. *Siemens Enterprise Communications UK-Security Solutions*, 7:53–58, 2008.

[18] Jung Yeon Hwang, Sungwook Eom, Ku-Young Chang, Pil Joong Lee, and Dae-Hun Nyang. Anonymity-based authenticated key agreement with full binding property. *IEEE Journal of Communications and Networks*, 18(2):190–200, 2016.

[19] Frank Stajano and Ross J. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Proc. of IWSP*, pages 172–194, 2000.

[20] M. Cagalj, S. Capkun, and J.-P. Hubaux. Key agreement in peer-to-peer wireless networks. *Proceedings of the IEEE*, 94(2):467–478, Feb. 2006.

[21] Sylvain Pasini and Serge Vaudenay. SAS-based authenticated key agreement. In *Proc. of PKC Conference*, volume 3958 of *LNCS*, pages 395 – 409, 2006.

[22] Sven Laur and Sylvain Pasini. SAS-based group authentication and key agreement protocols. In *Proc. of PKC Conference*, LNCS, pages 197–213, 2008.

[23] Toni Perkovic, Mario Cagalj, Toni Mastelic, Nitesh Saxena, and Dinko Begusic. Secure initialization of multiple constrained wireless devices for an unaided user. *IEEE Transactions on Mobile Computing*, 11(2):337–351, 2012.

[24] Jonathan M. McCune, Adrian Perrig, and Michael K. Reiter. Seeing-is-believing: Using camera phones for human-verifiable authentication. In *Proc. of Security and Privacy Symposium*, pages 110–124, 2005.

[25] R. Nithyanand, N. Saxena, G. Tsudik, and E. Uzun. Groupthink: Usability of secure group association for wireless devices. In *Proc. of ACM international conference on Ubiquitous computing*, pages 331–340, 2010.

[26] Arun Kumar, Nitesh Saxena, Gene Tsudik, and Ersin Uzun. Caveat eptor: A comparative study of secure device pairing methods. *Proc. of Percom*, pages 1–10, 2009.

[27] Long Hoang Nguyen and Andrew William Roscoe. Authentication protocols based on low-bandwidth unspoofable channels: a comparative survey. *Journal of Computer Security*, 19(1):139–201, 2011.

[28] Chia-Hsin Owen Chen, Chung-Wei Chen, Cynthia Kuo, Yan-Hao Lai, Jonathan M. McCune, Ahren Studer, Adrian Perrig, Bo-Yin Yang, and Tzong-Chen Wu. Gangs: gather, authenticate 'n group securely. In *Proc. of MOBICOM Conference*, pages 92–103, 2008.

[29] Yue-Hsun Lin, Ahren Studer, Hsu-Chin Hsiao, Jonathan M. McCune, King-Hang Wang, Maxwell Krohn, Phen-Lan Lin, Adrian Perrig, Hung-Min Sun, and Bo-Yin Yang. Spate: small-group pki-less authenticated trust establishment. In *Proc. of MOBISYS Conference*, pages 1–14, 2009.

[30] Michael T. Goodrich, Michael Sirivianos, John Solis, Gene Tsudik, and Ersin Uzun. Loud and clear: Human-verifiable authentication based on audio. In *Proc. of ICDCS Conference*, page 10, 2006.

[31] Srdjan Capkun, Mario Cagalj, RamKumar Rengaswamy, Ilias Tsigkogiannis, Jean-Pierre Hubaux, and Mani Srivastava. Integrity codes: Message integrity protection and authentication over insecure channels. *IEEE Transactions on Dependable and Secure Computing*, 5(4):208–223, 2008.

[32] Yantian Hou, Ming Li, and Joshua D. Guttman. Chorus: Scalable in-band trust establishment for multiple constrained devices over the insecure wireless channel. In *Proc. of the WiSec Conference*, pages 167–178, 2013.

[33] Shyamnath Gollakota, Nabeel Ahmed, Nickolai Zeldovich, and Dina Katabi. Secure in-band wireless pairing. In *Proc. of USENIX security symposium*, pages 1–16, 2011.

[34] Yantian Hou, Ming Li, Ruchir Chauhan, Ryan M. Gerdes, and Kai Zeng. Message integrity protection over wireless channel by countering signal cancellation: Theory and practice. In *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, pages 261–272, 2015.

[35] Yanjun Pan, Yantian Hou, Ming Li, Ryan M Gerdes, Kai Zeng, Md A Towfiq, and Bedri A Cetiner. Message integrity protection over wireless channel: Countering signal cancellation via channel randomization. *IEEE Transactions on Dependable and Secure Computing*, 2017.

[36] Junqi Zhang and Vijay Varadharajan. Wireless sensor network key management survey and taxonomy. *Journal of Network and Computer Applications*, 33(2):63–75, 2010.

[37] Namhi Kang. A first step towards security for internet of small things. *International Journal of Security and Its Applications*, 10(6):13–22, 2016.

[38] Laurent Eschenauer and Virgil D Gligor. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 41–47. ACM, 2002.

[39] Xiaohui Liang, Tianlong Yun, Ronald Peterson, and David Kotz. Lighttouch: Securely connecting wearables to ambient displays with user intent. In *Proc. of INFOCOM*, pages 1–9. IEEE, 2017.

[40] Wenlong Shen, Weisheng Hong, Xianghui Cao, Bo Yin, Devu Manikantan Shila, and Yu Cheng. Secure key establishment for device-to-device communications. In *Proc. of 2014 IEEE Global Communications Conference*, pages 336–340. IEEE, 2014.

[41] DaeHun Nyang, Aziz Mohaisen, and Jeonil Kang. Keylogging-resistant visual authentication protocols. *IEEE Transactions on Mobile Computing*, 13(11):2566–2579, 2014.

[42] International Air Transport Association et al. 2036 forecast reveals air passengers will nearly double to 7.8 billion, 2017.

[43] Michael G. Whitaker. *NextGen Works for America: Chief NextGen Officer Update to Congress*. Federal Aviation Administration, 2014. Pursuant to Section 204 of the FAA Modernization and Reform Act of 2012 (P.L. 112-95).

[44] Ian F Akyildiz, Brandon F Lo, and Ravikumar Balakrishnan. Cooperative spectrum sensing in cognitive radio networks: A survey. *Physical Comm.*, 4(1):40–62, 2011.

[45] Noor Al-Nakhala, Ryan Riley, and Tarek Elfouly. Distributed algorithms in wireless sensor networks: an approach for applying binary consensus in a real testbed. *Comp. Nets.*, 2015.

[46] IEEE 802.11 Working Group. IEEE 802.22 WRAN standards. `http://www.ieee802.org/22/`, 2011.

[47] Dimos V Dimarogonas, Emilio Frazzoli, and Karl H Johansson. Distributed event-triggered control for multi-agent systems. *IEEE Trans. on Aut. Cntrl.*, 57(5):1291–1297, 2012.

[48] Wi-Fi Alliance. Wi-fi alliance introduces security enhancements. *URL https://www. wi-fi. org/news-events/newsroom/wi-fi-allianceintroduces-security-enhancements, viitattu*, 27, 2018.

[49] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[50] Rongxing Lu and Zhenfu Cao. Simple three-party key exchange protocol. *Computers & Security*, 26(1):94–97, 2007.

[51] Christoph G Günther. An identity-based key-exchange protocol. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 29–37. Springer, 1989.

[52] Michael Steiner, Gene Tsudik, and Michael Waidner. Diffie-hellman key distribution extended to group communication. 1996.

[53] David Adrian, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, J Alex Halderman, Nadia Heninger, Drew Springall, Emmanuel Thomé, Luke Valenta, et al. Imperfect forward secrecy: How diffie-hellman fails in practice. In *Proc. of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 5–17. ACM, 2015.

[54] Matthias Wilhelm, Jens B Schmitt, and Vincent Lenders. Practical message manipulation attacks in ieee 802.15. 4 wireless networks. In *Proc. of Workshop MMB*, pages 29–31, 2012.

[55] Bruce Berg, Tyler Kaczmarek, Alfred Kobsa, and Gene Tsudik. Lights, camera, action! exploring effects of visual distractions on completion of security tasks. In *Proc. of International Conference on Applied Cryptography and Network Security*, pages 124–144. Springer, 2017.

[56] Ming Li, Shucheng Yu, Joshua D. Guttman, Wenjing Lou, and Kui Ren. Secure ad hoc trust initialization and key management in wireless body area networks. *ACM Trans. Sen. Netw.*, 9(2):18:1–18:35, Apr. 2013.

[57] Cynthia Kuo, Mark Luk, Rohit Negi, and Adrian Perrig. Message-in-a-bottle: user-friendly and secure key deployment for sensor nodes. In *Proc. of SenSys'07*, pages 233–246, 2007.

[58] Yee Wei Law, Giorgi Moniava, Zheng Gong, Pieter Hartel, and Marimuthu Palaniswami. Kalwen: A new practical and interoperable key management scheme for body sensor networks. *Security and communication networks*, 4(11):1309–1329, 2011.

[59] Debiao He, Sherali Zeadally, Neeraj Kumar, and Jong-Hyouk Lee. Anonymous authentication for wireless body area networks with provable security. *IEEE Systems Journal*, 11(4):2590–2601, 2017.

[60] Guanglou Zheng, Gengfa Fang, Rajan Shankaran, Mehmet A Orgun, Jie Zhou, Li Qiao, and Kashif Saleem. Multiple ecg fiducial points-based random binary sequence generation for securing wireless body area networks. *IEEE Journal of biomedical and health informatics*, 21(3):655–663, 2017.

[61] Nima Karimian, Paul A Wortman, and Fatemeh Tehranipoor. Evolving authentication design considerations for the internet of biometric things (iobt). In *Proc. of the Eleventh IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, page 10. ACM, 2016.

[62] Taha Belkhouja, Xiaojang Du, Amr Mohamed, Abdulla K Al-Ali, and Mohsen Guizani. New plain-text authentication secure scheme for implantable medical devices with remote control. In *Proc. of IEEE Global Communications Conference*, pages 1–5. IEEE, 2017.

[63] Chenglong Fu, Xiaojiang Du, Longfei Wu, and Xinwen Fu. Poks based low energy authentication scheme for implantable medical devices. *arXiv preprint arXiv:1803.09890*, 2018.

[64] Weitao Xu, Chitra Javali, Girish Revadigar, Chengwen Luo, Neil Bergmann, and Wen Hu. Gait-key: A gait-based shared secret key generation protocol for wearable devices. *ACM Transactions on Sensor Networks (TOSN)*, 13(1):6, 2017.

[65] Xiali Hei and Xiaojiang Du. Biometric-based two-level secure access control for implantable medical devices during emergencies. In *Proc. of 30th IEEE International Conference on Computer Communications*, pages 346 – 350, Shanghai, P.R.China, April 2011.

[66] Christian T Zenger, Jan Zimmer, Mario Pietersz, Jan-Felix Posielek, and Christof Paar. Exploiting the physical environment for securing the internet of things. In *Proc. of the 2015 New Security Paradigms Workshop*, pages 44–58. ACM, 2015.

[67] Dominik Schürmann and Stephan Sigg. Secure communication based on ambient audio. *IEEE Transactions on mobile computing*, 12(2):358–370, 2013.

[68] Eiji Hayashi, Sauvik Das, Shahriyar Amini, Jason Hong, and Ian Oakley. CASA: context-aware scalable authentication. In *Proc. of the Ninth Symposium on Usable Privacy and Security*, page 3. ACM, 2013.

[69] Markus Miettinen, N Asokan, Thien Duc Nguyen, Ahmad-Reza Sadeghi, and Majid Sobhani. Context-based zero-interaction pairing and key evolution for advanced personal devices. In *Proc. of the CCS Conference*, pages 880–891, 2014.

[70] Markus Miettinen, N Asokan, Thien Duc Nguyen, Ahmad-Reza Sadeghi, and Majid Sobhani. Context-based zero-interaction pairing and key evolution for advanced personal devices. In *Proc. of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 880–891. ACM, 2014.

[71] Kai Zeng, Kannan Govindan, and Prasant Mohapatra. Non-cryptographic authentication and identification in wireless networks. *Wireless Commun.*, 17:56–62, October 2010.

[72] Liang Cai, Kai Zeng, Hao Chen, and Prasant Mohapatra. Good neighbor: Ad hoc pairing of nearby wireless devices by multiple antennas. In *Proc. of Network and Distributed System Security Symposium*, 2011.

[73] Alex Varshavsky, Adin Scannell, Anthony LaMarca, and Eyal De Lara. Amigo: proximity-based authentication of mobile devices. In *Proc. of 9th international conference on Ubiquitous computing*, pages 253–270, 2007.

[74] Andre Kalamandeen, Adin Scannell, Eyal de Lara, Anmol Sheth, and Anthony LaMarca. Ensemble: cooperative proximity-based authentication. In *Proc. of MobiSys*, pages 331–344, 2010.

[75] Suhas Mathur, Robert Miller, Alexander Varshavsky, Wade Trappe, and Narayan Mandayam. Proximate: proximity-based secure pairing using ambient wireless signals. In *Proc. of MobiSys*, pages 211–224, 2011.

[76] T. J. Pierson, X. Liang, R. Peterson, and D. Kotz. Wanda: Securely introducing mobile devices. In *Proc. of INFOCOM*, pages 1–9, 2016.

[77] Jiansong Zhang, Zeyu Wang, Zhice Yang, and Qian Zhang. Proximity based iot device authentication. In *Proc. of INFOCOM*, 2017.

[78] Wei Cheng, Kefeng Tan, Victor Omwando, Jindan Zhu, and Prasant Mohapatra. Rss-ratio for enhancing performance of rss-based applications. In *Proc. of 2013 INFOCOM*, pages 3075–3083. IEEE, 2013.

[79] S. Brands and D. Chaum. Distance-bounding protocols. In *Proc. of Advances in Cryptology EUROCRYPT*, pages 344–359. Springer, 1994.

[80] K.B. Rasmussen and S. Capkun. Realization of rf distance bounding. In *Proc. of USENIX Security Symposium*, 2010.

[81] K.B. Rasmussen, C. Castelluccia, T.S. Heydt-Benjamin, and S. Capkun. Proximity-based access control for implantable medical devices. In *Proc. of 16th ACM conference on Computer and communications security*, pages 410–419, 2009.

[82] N. Patwari and S.K. Kasera. Robust location distinction using temporal link signatures. In *Proc. of 13th annual ACM international conference on Mobile computing and networking*, pages 111–122, 2007.

[83] Di Ma, Nitesh Saxena, Tuo Xiang, and Yan Zhu. Location-aware and safer cards: enhancing rfid security and privacy via location sensing. *IEEE transactions on dependable and secure computing*, 10(2):57–69, 2013.

[84] Thien Duc Nguyen, Samuel Marchal, Markus Miettinen, Minh Hoang Dang, N Asokan, and Ahmad-Reza Sadeghi. DÏot: A crowdsourced self-learning approach for detecting compromised iot devices. *arXiv preprint arXiv:1804.07474*, 2018.

[85] Pieter Robyns, Eduard Marin, Wim Lamotte, Peter Quax, Dave Singelée, and Bart Preneel. Physical-layer fingerprinting of LoRa devices using supervised and zero-shot learning. In *Proc. of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2017.

[86] Muhammad Naveed Aman, Kee Chaing Chua, and Biplab Sikdar. Mutual authentication in iot systems using physical unclonable functions. *IEEE Internet of Things Journal*, 4(5):1327–1340, 2017.

[87] Lanxiang Chen. A framework to enhance security of physically unclonable functions using chaotic circuits. *Physics Letters A*, 382(18):1195–1201, 2018.

[88] Urbi Chatterjee, Vidya Govindan, Rajat Sadhukhan, Debdeep Mukhopadhyay, Rajat Subhra Chakraborty, Debashis Mahata, and Mukesh M Prabhu. Building PUF based authentication and key exchange protocol for IoT without explicit CRPs in verifier database. *IEEE Transactions on Dependable and Secure Computing*, 2018.

[89] Andrea M Tonello, Nicola Laurenti, and Silvano Pupolin. Analysis of the uplink of an asynchronous multi-user dmt ofdma system impaired by time offsets, frequency offsets, and multi-path fading. In *Vehicular Technology Conference,*

2000. IEEE-VTS Fall VTC 2000. 52nd, volume 3, pages 1094–1099. IEEE, 2000.

[90] Amritha Sampath and C Tripti. Synchronization in distributed systems. In *Advances in Computing and Information Technology*, pages 417–424. Springer, 2012.

[91] Theodore S Rappaport et al. *Wireless communications: principles and practice*, volume 2. Prentice Hall PTR New Jersey, 1996.

[92] Dirk Balfanz, Diana K Smetters, Paul Stewart, and H Chi Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *Proc. of NDSS Symposium*, 2002.

[93] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In *Proc. of CRYPTO*, pages 531–545. Springer, 2000.

[94] Danny Dolev and Andrew Yao. On the security of public key protocols. *IEEE Transactions on information theory*, 29(2):198–208, 1983.

[95] Maja Stella, Mladen Russo, and D Begusic. Location determination in indoor environment based on rss fingerprinting and artificial neural network. In *Proc. of ConTel Conference*, pages 301–306, 2007.

[96] Oktay Ureten and Nur Serinken. Wireless security through rf fingerprinting. *Canadian Journal of Electrical and Computer Engineering*, 32(1):27–33, 2007.

[97] Chahé Nerguizian, Charles Despins, and Sofiène Affès. Geolocation in mines with an impulse response fingerprinting technique and neural networks. *IEEE Transactions on Wireless Communications*, 5(3):603–611, 2006.

[98] Vladimir Brik, Suman Banerjee, Marco Gruteser, and Sangho Oh. Wireless device identification with radiometric signatures. In *Proc. of Mobicom*, pages 116–127, 2008.

[99] Jeyanthi Hall, Michel Barbeau, and Evangelos Kranakis. Enhancing intrusion detection in wireless networks using radio frequency fingerprinting. In *Proc. of Communications, Internet, and Information Technology*, pages 201–206, 2004.

[100] Heikki Laitinen, Jaakko Lahteenmaki, and Tero Nordstrom. Database correlation method for gsm location. In *Proc. of Vehicular Technology Conference*, volume 4, pages 2504–2508, 2001.

[101] Bocan Hu, Yan. Zhang, and Loukas Lazos. PHYVOS: Physical layer voting for secure and fast cooperation. In *Proc. of IEEE Conference on Communications and Networks Security*, 2015.

[102] Victor Boyko, Philip MacKenzie, and Sarvar Patel. Provably secure password-authenticated key exchange using diffie-hellman. In *Proc. of Eurocrypt*, pages 156–171, 2000.

[103] Douglas R. Stinson. *Cryptography: theory and practice.* CRC press, 2005.

[104] Victor Rabinovich and Nikolai Alexandrov. Typical array geometries and basic beam steering methods. In *Antenna Arrays and Automotive Applications*, pages 23–54. Springer, 2013.

[105] Hubregt J Visser. *Array and phased array antenna basics.* John Wiley & Sons, 2006.

[106] Hiroshi Akima. A new method of interpolation and smooth curve fitting based on local procedures. *Journal of the ACM (JACM)*, 17(4):589–602, 1970.

[107] Shahab Mirzadeh, Haitham S Cruickshank, and Rahim Tafazolli. Secure device pairing: A survey. *IEEE Communications Surveys and Tutorials*, 16(1):17–40, 2014.

[108] IEEE standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements - part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pages 1–3534, Dec 2016.

[109] IEEE standard for low-rate wireless networks - amendment 5: Enabling/updating the use of regional sub-ghz bands. *IEEE Std 802.15.4v-2017 (Amendment to IEEE Std 802.15.4-2015, as amended by IEEE Std 802.15.4n-2016, IEEE Std 802.15.4q-2016, IEEE Std 802.15.4u-2016, and IEEE Std 802.15.4t-2017)*, pages 1–35, June 2017.

[110] IEEE standard for information technology–telecommunications and information exchange between systems - local and metropolitan area networks–specific requirements - part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 2: Sub 1 ghz license exempt operation. *IEEE Std 802.11ah-2016 (Amendment to IEEE Std 802.11-2016, as amended by IEEE Std 802.11ai-2016)*, pages 1–594, April 2017.

[111] IEEE standard for information technology– telecommunications and information exchange between systemslocal and metropolitan area networks– specific

requirements–part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications–amendment 4: Enhancements for very high throughput for operation in bands below 6 ghz. *IEEE Std 802.11ac-2013 (Amendment to IEEE Std 802.11-2012, as amended by IEEE Std 802.11ae-2012, IEEE Std 802.11aa-2012, and IEEE Std 802.11ad-2012)*, pages 1–425, Dec 2013.

[112] Emmanuel Bresson, Olivier Chevassut, David Pointcheval, and Jean-Jacques Quisquater. Provably authenticated group diffie-hellman key exchange. In *Proceedings of the 8th ACM conference on Computer and Communications Security*, pages 255–264. ACM, 2001.

[113] ILYA I BOGDANOV. Two theorems on the focus-sharing ellipses: a three-dimensional view. *Journal of Classical Geometry Volume 1 (2012)*, 1:1, 2012.

[114] Aveek Dutta, Dola Saha, Dirk Grunwald, and Douglas Sicker. SMACK: a SMart ACKnowledgment scheme for broadcast messages in wireless networks. In *ACM SIGCOMM Computer Communication Review*, volume 39, pages 15–26. ACM, 2009.

[115] Wi-Fi Alliance. Wi-fi protected setup specification. *WiFi Alliance Document*, 23, 2007.

[116] Wei Peng, Yubai Li, Huan Li, and Bolong Wen. A novel high-sensitivity ADS-B receiver based on RF direct logarithmic detecting. In *Proc. of the International Conference on Computer Application and System Modeling*, 2012.

[117] George Wright. NAV CANADA implements ADS-B. In *Proc. of the Integrated Communications, Navigation and Surveillance Conference*, pages 1–9, 2009.

[118] Fabrice Kunzi and R John Hansman. ADS-B benefits to general aviation and barriers to implementation. `http://hdl.handle.net/1721.1/63130`, 2011.

[119] Joonsang Baek, Young-ji Byon, Eman Hableel, and Mahmoud Al-Qutayri. Making air traffic surveillance more reliable: A new authentication framework for automatic dependent surveillance-broadcast (ADS-B) based on online/offline identity-based signature. *Security and Communication Networks*, 2014.

[120] Alexander E Smith. Method and apparatus for improving ADS-B security, 2008. US Patent 7,423,590.

[121] Matthias Schäfer, Vincent Lenders, and Ivan Martinovic. Experimental analysis of attacks on next generation air traffic communication. In *Proc. of the Applied Cryptography and Network Security Conference*, pages 253–271, 2013.

[122] Martin Strohmeier, Vincent Lenders, and Ivan Martinovic. Security of ADS-B: State of the art and beyond. *arXiv preprint arXiv:1307.3664*, 2013.

[123] Kyle D Wesson, Todd E Humphreys, and Brian L Evans. Can cryptography secure next generation air traffic surveillance? *IEEE Security and Privacy Magazine*, 2014.

[124] Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced oil and vinegar signature schemes. In *Proc. of EUROCRYPT*, pages 206–222. Springer, 1999.

[125] Federal Aviation Administration. *Automatic Dependent Surveillance Broadcast (ADS-B) Out Performance Requirements to Support Air Traffic Control (ATC) Service; Final Rule.* 14 CFR Part 91 Federal Register, 2010. 75(103).

[126] Mark D Austin and GL Stüber. Velocity adaptive handoff algorithms for microcellular systems. *IEEE Transactions on Vehicular Technology*, 43(3):549–561, 1994.

[127] Michael J Chu and Wayne E Stark. Effect of mobile velocity on communications in fading channels. *IEEE Transactions on Vehicular Technology*, 49(1):202–210, 2000.

[128] Shengquan Hu, Tugay Eyceoz, Alexandra Duel-Hallen, and Hans Hallen. Transmitter antenna diversity and adaptive signaling using long range prediction for fast fading ds/cdma mobile radio channels. In *Proc. of the Wireless Communications and Networking Conference*, pages 824–828, 1999.

[129] Jack M Holtzman and Ashwin Sampath. Adaptive averaging methodology for handoffs in cellular systems. *Vehicular Technology*, 44(1):59–66, 1995.

[130] Goohyun Park, Daesik Hong, and Changeon Kang. Level crossing rate estimation with Doppler adaptive noise suppression technique in frequency domain. In *Proc. of the VTC Conference*, volume 2, pages 1192–1195, 2003.

[131] Cihan Tepedelenlioğlu and Georgios B Giannakis. On velocity estimation and correlation properties of narrow-band mobile communication channels. *IEEE Transactions on Vehicular Technology*, 50(4):1039–1052, 2001.

[132] Weidong Xiang, Paul Richardson, and Jinhua Guo. Introduction and preliminary experimental results of wireless access for vehicular environments (WAVE) systems. In *Proc. of the Mobile and Ubiquitous Systems-Workshop*, pages 1–8, 2006.

[133] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial & Applied Mathematics*, 11(2):431–441, 1963.

[134] Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *Advances in Cryptology*, pages 392–407, 2000.

[135] Ruidong Chen, Chengxiang Si, Haomiao Yang, and Xiaosong Zhang. ADS-B data authentication based on AH protocol. In *Proc. of the IEEE International Conference on Dependable, Autonomic and Secure Computing*, pages 21–24, 2013.

[136] Jimmy Krozel, Dominick Andrisani, Mohammad A Ayoubi, Takayuki Hoshizaki, and Chris Schwalm. Aircraft ADS-B data integrity check. In *Proc. of the AIAA Aviation, Technology, Integration, and Operations Conference*, 2004.

[137] Hariharan Rahul, Haitham Hassanieh, and Dina Katabi. SourceSync: a distributed wireless architecture for exploiting sender diversity. *ACM SIGCOMM Comp. Comm. Rev.*, 41(4):171–182, 2011.

[138] Aveek Dutta, Dola Saha, Dirk Grunwald, and Douglas Sicker. SMACK: a SMart ACKnowledgment scheme for broadcast messages in wireless networks. *ACM SIGCOMM Comp. Comm. Rev.*, 39(4):15–26, 2009.

[139] Krishna Sampigethaya and Radha Poovendran. A framework and taxonomy for comparison of electronic voting schemes. *Computers & Security*, 25(2):137–153, 2006.

[140] Richard van Nee and Ramjee Prasad. *OFDM for wireless multimedia communications*. Artech House, Inc., 2000.

[141] Tao Jin, Guevara Noubir, and Bishal Thapa. Zero pre-shared secret key establishment in the presence of jammers. In *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, pages 219–228. ACM, 2009.

[142] Yao Liu, Peng Ning, Huaiyu Dai, and An Liu. Randomized differential dsss: Jamming-resistant wireless broadcast communication. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[143] Srdjan Capkun, Mario Cagalj, RamKumar Rengaswamy, Ilias Tsigkogiannis, Jean-Pierre Hubaux, and Mani Srivastava. Integrity codes: Message integrity protection and authentication over insecure channels. *Dependable and Secure Computing, IEEE Transactions on*, 5(4):208–223, 2008.

[144] Suhas Mathur, Wade Trappe, Narayan Mandayam, Chunxuan Ye, and Alex Reznik. Radio-telepathy: extracting a secret key from an unauthenticated

wireless channel. In *Proc. of the MOBICOM Conf.*, pages 128–139. ACM, 2008.

[145] Dinesh Bharadia, Emily McMilin, and Sachin Katti. Full duplex radios. In *Proc. of the SIGCOMM Computer Communication Review*, pages 375–386. ACM, 2013.

[146] Achaleshwar Sahai, Gaurav Patel, and Ashutosh Sabharwal. Pushing the limits of full-duplex: Design and real-time implementation. *arXiv preprint arXiv:1107.0607*, 2011.

[147] Zhaowu Zhan, Guillaume Villemaud, and Jean-Marie Gorce. Design and evaluation of a wideband full-duplex ofdm system based on aasic. In *IEEE Personal Indoor and Mobile Radio Communications (PIMRC), 2013*, pages 68–72, 2013.

[148] Jeffrey G Andrews, Arunabha Ghosh, and Rias Muhamed. *Fundamentals of WiMAX: understanding broadband wireless networking.* Pearson Education, 2007.

[149] Hanif Rahbari, Marwan Krunz, and Loukas Lazos. Swift jamming attack on frequency offset estimation: The achilles? heel of OFDM systems. *IEEE Transactions on Mobile Computing*, 15(5):1264–1278, 2016.

[150] Hanif Rahbari, Marwan Krunz, and Loukas Lazos. Security vulnerability and countermeasures of frequency offset correction in 802.11 a systems. In *INFOCOM, 2014 Proceedings IEEE*, pages 1015–1023. IEEE, 2014.

[151] LitePoint. Practical manufacturing testing of 802.11 OFDM wireless devices. `http://www.litepoint.com/whitepaper/Testing\%20802.11\%20OFDM\%20Wireless\%20Devices_WhitePaper.pdf`, 2012.

[152] MATLAB. *version 8.6.0 (R2015b).* The MathWorks Inc., Natick, Massachusetts, 2015.

[153] John Von Neumann. Probabilistic logics and the synthesis of reliable organisms from unreliable components. *Aut. studies*, 34:43–98, 1956.

[154] Michael Barborak, Anton Dahbura, and Miroslaw Malek. The consensus problem in fault-tolerant computing. *ACM Comp. Surveys*, 25(2):171–220, 1993.

[155] Gregory Levitin. Weighted voting systems: reliability versus rapidity. *Reliability Engineering & System Safety*, 89(2):177–184, 2005.

[156] Gregory Levitin and Anatoly Lisnianski. Reliability optimization for weighted voting system. *Reliability engineering & system safety*, 71(2):131–138, 2001.

[157] Daniel Barbara and Hector Garcia-Molina. The reliability of voting mechanisms. *IEEE Trans. Computers*, 36(10):1197–1208, 1987.

[158] Kevin Kwiat, Alan Taylor, William Zwicker, Daniel Hill, Sean Wetzonis, and Shangping Ren. Analysis of binary voting algorithms for use in fault-tolerant and secure computing. In *Computer Engineering and Systems (ICCES), 2010 International Conference on*, pages 269–273. IEEE, 2010.

[159] Li Wang, Zheng Li, Shangping Ren, and Kevin Kwiaty. Optimal voting strategy against rational attackers. In *Risk and Security of Internet and Systems (CRiSIS), 2011 6th International Conference on*, pages 1–8. IEEE, 2011.

[160] W. Kim, K. Mechitov, Jeung Choi, and S. Ham. On target tracking with binary proximity sensors. In *Proc. of the IPSN*, pages 301–308, 2005.

[161] Mengxia Zhu, Song Ding, Qishi Wu, R. R. Brooks, N. S. V. Rao, and S. S. Iyengar. Fusion of threshold rules for target detection in wireless sensor networks. *ACM Trans. on Sens. Nets.*, 6(2):181–187, 2010.

[162] Loukas Lazos and Radha Poovendran. SeRLoc: robust localization for wireless sensor networks. *ACM Trans. on Sens. Nets.*, 1(1):73–100, 2005.

[163] Xuanwen Luo, Ming Dong, and Yinlun Huang. On distributed fault-tolerant detection in wireless sensor networks. *IEEE Trans. on Comp.*, 55(1):58–70, 2006.

[164] E Ould-Ahmed-Vall, Bonnie Heck Ferri, and George F Riley. Distributed fault-tolerance for event detection using heterogeneous wireless sensor networks. *IEEE Trans. on Mob. Comp.*, 11(12):1994–2007, 2012.