

## ABSTRACT

Title of Dissertation:      PHYSICAL LAYER AUTHENTICATION

Paul Yu, Doctor of Philosophy, 2008

Dissertation directed by: Professor John S. Baras  
Department of Electrical and Computer Engineering

A fundamental problem in security is authentication: namely, how to verify the identity of another party. Without this verification, the ideas of privacy and integrity are moot. Modern authentication techniques use cryptographic operations that secure the system against adversaries that do not have tremendous amounts of computation and memory. However, when the abilities of the adversary increase, such authentication paradigms become more susceptible to defeat. With the greater threat of defeat, the secret authentication keys must be replaced more often. Unfortunately, the popular key replacement algorithms typically rely on either third parties or on non-trivial computational ability. In this thesis we attack these two aspects of the authentication problem by presenting novel methods for authentication and key replacement in wireless environments.

We describe how to exploit the randomness of the physical layer to hide the authentication from adversaries. Typically, no effort is made to hide the authentication - it is sent in plain view of friend and foe alike. The proposed technique reveals significantly less key information than traditional authentication methods and can increase the data throughput of the system. We define metrics to quantify the performance of the proposed authentication system and use them to study the associated tradeoffs. A software radio implementation is then presented to demonstrate the feasibility of the proposed scheme.

Finally, we consider how secret keys can be replaced in an efficient manner. We describe a novel method of key replacement and generation that, unlike other methods, requires no additional message exchanges after initialization and yet generates highly random keys. As an added benefit, the method is shown to be extremely lightweight in terms of computation and memory requirements.

PHYSICAL LAYER AUTHENTICATION

by

Paul Yu

Dissertation submitted to the Faculty of the Graduate School of the  
University of Maryland, College Park in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
2008

Advisory Committee:

Professor John S. Baras, Chairman/Advisor  
Professor Prakash Narayan  
Professor Şennur Ulukuş  
Professor Michel Cukier  
Professor Lawrence C. Washington

© Copyright by  
Paul Yu  
2008

## ACKNOWLEDGEMENTS

This dissertation could not have been written without Dr. John Baras, who not only served as my academic advisor but also as my inspiration, continually encouraging and supporting me through my studies. I thank him for always being an inspiring and dynamic force who gets me excited about pushing the envelope and delivering tangible results. I deeply appreciate him for guiding me towards interesting problems while allowing me the freedom to explore topics of my own interest. His depth and breadth of experience and expertise were freely shared with me, and for that I am truly enriched.

I would also like to express my tremendous gratitude to Dr. Brian Sadler, who together with Dr. Baras, mentored me in research and in life. Through his patient guidance he taught me better ways of formulating problems, finding solutions, and expressing ideas. Always willing to give detailed feedback, he spent countless hours helping me improve my work. My skills as a researcher have greatly benefited from his selfless mentorship.

I would like to express my sincere thanks to Professors Prakash Narayan, Şennur Ulukuş, Michel Cukier, and Lawrence Washington for spending their valuable time reading my thesis, taking part in my dissertation defense committee, and providing me with helpful advice, comments, and suggestions.

I am grateful to my many friends and colleagues at the university who have made my experience here not only memorable but enjoyable as well. To Althia Kirlew and Kim Edwards I am indebted for their kindness and generous assistance in a variety of matters in which

they are far more capable than I. This work was supported by the CTA contract DAAD 19-01-2-001.

My family has always been around me, and it is their unconditional love and support that enabled me to enjoy my studies. I would like to thank my parents and my three wonderful sisters May, Rebecca, and Joyce, who continually remind me of what true beauty is. Finally, I would like to thank my beloved Rosa for her endless love, support, enthusiasm, and patience. Her unwavering optimism and unceasing encouragement continue to motivate me each day. Their belief in me motivated me to reach higher and work harder. This thesis is dedicated to them.

# TABLE OF CONTENTS

<b>List of Tables</b>	<b>vii</b>
<b>List of Figures</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Problem . . . . .	1
1.2 Motivating Examples . . . . .	2
1.2.1 Cryptography is Vulnerable to Advances in Adversary Ability	2
1.2.2 Key Replacement is Not Cheap . . . . .	4
1.3 Approach . . . . .	5
1.3.1 Authentication Method . . . . .	5
1.3.2 Key Replacement . . . . .	7
1.4 Contributions and Roadmap of This Thesis . . . . .	9
<b>2 Preliminaries</b>	<b>11</b>
2.1 Overview . . . . .	11
2.2 Cast of Characters . . . . .	12
2.2.1 Scenario . . . . .	12
2.2.2 Channel Assumptions . . . . .	13
2.3 Authentication . . . . .	14
2.3.1 Factors of Authentication . . . . .	14
2.3.2 Common Implementations . . . . .	15
2.3.3 Attacks . . . . .	16
2.3.4 Properties of the Authentication Tag . . . . .	18
2.4 Wireless Channels . . . . .	19
2.4.1 Channel Models . . . . .	19
2.4.2 Uncertainties . . . . .	22
2.4.3 Capacity vs. Rate . . . . .	25
2.5 Evaluation Metrics . . . . .	27
2.5.1 Hypothesis Testing . . . . .	28
2.5.2 Stealth . . . . .	29
2.5.3 Robustness . . . . .	32
2.5.4 Security . . . . .	33

2.6	Related Work . . . . .	40
2.6.1	Cryptographic and Information Theoretic Security . . . . .	40
2.6.2	Unconditionally Secure Authentication . . . . .	43
2.7	Conclusion . . . . .	45
<b>3</b>	<b>Physical Layer Authentication Framework</b>	<b>46</b>
3.1	Overview of Contributions . . . . .	46
3.2	Framework . . . . .	47
3.2.1	Signal Model . . . . .	47
3.2.2	Channel Model and Estimation . . . . .	48
3.2.3	Message Recovery . . . . .	50
3.2.4	Tag Detection . . . . .	54
3.2.5	Superposition Methods . . . . .	60
3.3	Metric Evaluation . . . . .	62
3.3.1	Stealth . . . . .	63
3.3.2	Robustness . . . . .	66
3.3.3	Security . . . . .	68
3.4	Extension to Fast-Fading Channels . . . . .	70
3.4.1	Channel Model . . . . .	71
3.4.2	Channel Estimation . . . . .	71
3.4.3	Message Recovery . . . . .	74
3.4.4	Authentication . . . . .	75
3.4.5	Example and Results . . . . .	75
3.5	Conclusion . . . . .	77
<b>4</b>	<b>Multi-Carrier Authentication</b>	<b>79</b>
4.1	Overview of Contributions . . . . .	79
4.2	Framework . . . . .	80
4.2.1	Signal Model . . . . .	81
4.2.2	Channel Model and Estimation . . . . .	84
4.2.3	Message Recovery . . . . .	88
4.2.4	Tag Recovery . . . . .	92
4.2.5	Practical Considerations . . . . .	97
4.3	Metric Evaluation . . . . .	100
4.3.1	Stealth . . . . .	100
4.3.2	Robustness . . . . .	103
4.3.3	Security . . . . .	105
4.4	Power Allocation . . . . .	111
4.4.1	Strategies . . . . .	111
4.4.2	Capacity . . . . .	117
4.4.3	Authentication Metrics . . . . .	117
4.5	Conclusion . . . . .	123

<b>5</b>	<b>Experimental Results</b>	<b>125</b>
5.1	Overview of Contributions . . . . .	125
5.2	Experiment Setup . . . . .	126
5.2.1	Hardware Capabilities . . . . .	127
5.2.2	Software Design . . . . .	130
5.3	Testing Procedure and Results . . . . .	132
5.3.1	Stealth . . . . .	134
5.3.2	Presence . . . . .	136
5.3.3	Robustness . . . . .	139
5.3.4	Security . . . . .	140
5.4	Conclusion . . . . .	141
<b>6</b>	<b>Markov Key Replacement</b>	<b>142</b>
6.1	Overview of Contributions . . . . .	142
6.2	The Key Replacement Problem . . . . .	143
6.2.1	Example . . . . .	144
6.2.2	Key Replacement Paradigms . . . . .	144
6.3	Markov Key Replacement Method . . . . .	145
6.3.1	Overview with example . . . . .	145
6.3.2	Message model . . . . .	146
6.3.3	Markov model . . . . .	147
6.3.4	Key Replacement Algorithm . . . . .	149
6.3.5	Complexity and Costs . . . . .	150
6.3.6	Imperfect Key Recovery . . . . .	151
6.4	Markov Model . . . . .	155
6.4.1	Specification . . . . .	155
6.4.2	Construction . . . . .	156
6.4.3	Model Properties . . . . .	157
6.4.4	Codebook Properties . . . . .	165
6.4.5	Entropy Rate . . . . .	169
6.5	Applications . . . . .	170
6.5.1	Cryptography . . . . .	170
6.5.2	Physical Layer Authentication . . . . .	170
6.5.3	Frequency Hopping Communications . . . . .	171
6.6	Related Work . . . . .	172
6.7	Conclusion . . . . .	173
<b>7</b>	<b>Future Directions</b>	<b>174</b>
	<b>Bibliography</b>	<b>176</b>

## LIST OF TABLES

3.1	Simulation parameters for the single carrier, Rayleigh block fading case . . . . .	63
3.2	Simulation parameters for the single carrier, Gauss-Markov channel case . . . . .	76
4.1	Simulation parameters for the multi-carrier, no CSI case . . . . .	101
4.2	Simulation parameters for the multi-carrier, perfect CSI case . . . . .	119
5.1	Authentication Probability . . . . .	139
6.1	Size of the GCC . . . . .	163

## LIST OF FIGURES

2.1	Scenario with Alice, Bob, Carol, and Eve. . . . .	13
2.2	Time division multiplexed (TDM) versus superimposed (SI) tags .	26
2.3	Rate of superimposed BPSK . . . . .	27
2.4	Pigeonhole principle. a) If another key is added, the number of keys will exceed the number of messages, there will be multiple keys that map at least 1 pair of (message,tag). b) A similar situation arises when the number of keys exceeds the number of tags. . . .	37
2.5	Wiretap Channel . . . . .	42
3.1	Block diagram of aware receiver. . . . .	50
3.2	4/16 hierarchical QAM constellation. The 16 white circles indicate the constellation points; the black circles indicate the original 4-QAM constellation used for the message symbols. . . . .	51
3.3	Message BER are compared for hierchical 4/16 QAM constellation versus normally distributed tags. The error floor for the Gaussian tags is determined by the message/tag power allocation $\rho^s$ . . . .	61
3.4	The wavelet tiling of the time-frequency plane. The wavelet basis yields a multi-resolution view of the signal that trades time resolution for frequency resolution. . . . .	62
3.5	Stealth: KL Distance between tagged and untagged signals for various message powers in AWGN. Higher message power makes the tag more hidden and hence leads to better stealth. . . . .	64
3.6	Stealth: Frame outage probabilities for various BCH codes in the Rayleigh fading channel. Higher message power admits lower frame outage probability and hence better stealth. . . . .	65
3.7	Robustness: KL Distance between correct and incorrect tags for various message powers in AWGN. Lower message power makes the tag more easily detected to Bob and hence better robustness. . . . .	66
3.8	Robustness: Authentication outage probabilities for various message powers in the Rayleigh fading channel. Lower message power admits lower authentication outage and hence better robustness. . . . .	67
3.9	Robustness: Authentication outage probabilities for various tag lengths in the Rayleigh fading channel. Longer frames admit higher authentication probability and hence better robustness. . . . .	68

3.10	Security: Equivocation of tags for various message powers in the Rayleigh fading channel. Higher message power leads to higher equivocation and hence better security. . . . .	70
3.11	TDM Pilot Placement. . . . .	72
3.12	AR-1 Stealth: Frame outage probabilities for various BCH codes in the AR-1 channel with $a = 0.99$ . Higher message power admits lower frame outage probability and hence better stealth. . . . .	76
3.13	AR-1 Robustness: Frame authentication probabilities for various message powers in the AR-1 channel with $a = 0.99$ . Lower message power admits higher authentication probability and hence better robustness. . . . .	77
3.14	AR-1 Security: Equivocation of tag symbols for various message powers in the AR-1 channel with $a = 0.99$ . Higher message power leads to higher equivocation and hence better security. . . . .	78
4.1	Example tag placements with $N = 4, N^f = 8$ and tag spread $N^t = 2$ . a) tag on specific carriers only, b) general tag placement. . . . .	83
4.2	Distribution of test statistic when $N = 32, N^t = 8, N^f = 4$ . Kurtosis values are $0.69(n = 8), 0.37(n = 16), 0.19(n = 32)$ . Increasing $n$ tightens the distribution about its mean, improving tag detection performance of the receiver. . . . .	97
4.3	Each tag symbol becomes less powerful as the tag is spread out over more symbols. . . . .	98
4.4	Stealth: KL distance between tagged and untagged signals for various $N_t$ in AWGN. Increased tag spread (higher $N_t$ ) decreases the KL distance and hence leads to better stealth. . . . .	102
4.5	Stealth: BER for various $N_t$ in AWGN. Increased tag spread (higher $N_t$ ) decreases the BER for a fixed message power, though the effect diminishes as the message power increases. . . . .	103
4.6	Robustness: Authentication probabilities for various tag spreads $N_t$ in the Rayleigh fading channel. Spreading the tag over more symbols yields higher authentication probability and hence better robustness. . . . .	104
4.7	Robustness: Authentication probabilities for various frequency offsets $\varepsilon$ in the Rayleigh fading channel. Small offsets inflict a manageable decrease in authentication performance. . . . .	105
4.8	Security: Equivocation of the authentication tags for various tag spreads $N_t$ in the Rayleigh fading channel. Spreading the tag over more symbols yields higher tag equivocation and hence better security. . . . .	107

4.9	Security: Equivocation of the authentication tags for various frequency offsets $\varepsilon$ in the Rayleigh fading channel. Increased $\varepsilon$ (worse ICI) yields higher tag equivocation and hence better security. . . .	108
4.10	Security: Key information gained when tag positions are unknown in the Rayleigh fading channel. False alarm probability is zero. The higher the miss detection probability the better the security. .	110
4.11	Power allocation strategies. Base bars represent noise power on the carriers, white bars represent message power, and lightly shaded bars represent tag power. Power allocation is 80% message and 20% tag ( $P^s = 0.8, P^t = 0.2$ ). . . . .	112
4.12	Message Capacity . . . . .	118
4.13	Tag Capacity . . . . .	118
4.14	Throughput for various strategies. Frame length = 32 symbols. Average SNR = 9 dB. . . . .	120
4.15	Stealth for various strategies. Frame length = 32 symbols. Average SNR = 9 dB. . . . .	121
4.16	Robustness for various strategies. Frame length = 32 symbols. Average SNR = 9 dB. False alarm probability $\alpha = 0.01$ . . . . .	122
4.17	Tag equivocation for various strategies. Frame length = 32 symbols. Average SNR = 9 dB. False alarm probability $\alpha = 0.01$ . . .	123
5.1	A fundamental concept of software defined radios is the placement of software as close as possible to the antennae. Only an analog-to-digital converter (ADC) separates the software from the antenna in the receive path (a), while a DAC is present in the transmit path (b). . . . .	126
5.2	An overview of the hardware setup: the laptop is connected via USB to the USRP. The USRP consists of an FPGA responsible for up/down conversions, ADCs and DACs, and various plug-in daughterboards. . . . .	128
5.3	Transmitter signal path. Unmodified processing blocks are grayed out; modifications are darkened. . . . .	131
5.4	Packet format. Note that the tag has non-null information coincident with the packet payload; no other portion fo the packet is modified by the superposition. . . . .	131
5.5	Receiver signal path. Unmodified processing blocks are grayed out; modifications are darkened. . . . .	132
5.6	The test scenarios and data sets are used to evaluate the authentication system. The data collected in each test scenario is used to compute the stealth, robustness, or stealth metrics. . . . .	134

5.7	The packet error rate for various sample runs versus the power of the authentication signal. At low authentication powers, no significant deviation from the baseline packet error rate was observed. Each line represents a different test run. . . . .	135
5.8	The observed SNR of tagged and untagged signals for a few consecutive packets. The majority of the packet SNR in three cases fall inside the 95% confidence interval for no authentication present in the signal; in this snapshot most of those that fall outside of it are actually false alarms. . . . .	137
5.9	The observed CDF of the estimated noise for various authentication powers over thousands of packets. Larger authentication powers deviate more from the baseline CDF. . . . .	138
5.10	Test statistic histograms for various length payloads. Longer payloads yield better signal separation and hence better authentication performance. . . . .	140
6.1	Simple Markov Model when $K = 4$ and $b = 2$ . The states and possible transitions are shown in (a) while a corresponding probability matrix is shown in (b). Note that $\mathbf{A}$ is sparse. . . . .	146
6.2	Probability of choosing the incorrect key for various raw detection key probabilities $p$ when $\alpha = 10^{-7}$ . Extending the key replacement over multiple messages encrypted with the same key improves the detection probability. Lower $p$ require more observations for good detection probability. . . . .	154
6.3	Number of key transitions before failure for various confidences. . . . .	155
6.4	A fully specified Markov Model with 32-bit keys, $K = 2^{32}$ , branching factor $d = 2$ . Requires at least 1 gigabyte to store. . . . .	156
6.5	Construction of a random access Markov Model. . . . .	157
6.6	a) Small reachable subspace b) Periodic . . . . .	158
6.7	An example bowtie digraph connected to the giant connected component $\mathcal{G}$ . $\mathcal{G}^+$ leads into $\mathcal{G}$ , $\mathcal{G}^-$ emanates from $\mathcal{G}$ . . . . .	164
6.8	Entropy of next key. $K = 2^{64}$ . . . . .	167
6.9	Distribution of stationary probabilities are well approximated with Rayleigh distribution for small $d$ . . . . .	168
6.10	Distribution of stationary probabilities are well approximated with Gaussian distribution for large $d$ . . . . .	168
6.11	Entropy of Model given current key. $K = 2^{10}$ . . . . .	169

# Chapter 1

## Introduction

### 1.1 Problem

The first step towards secure communication is authentication. Unless you know and trust the person you are talking with, it is clearly unwise to share secrets with her, no matter how secure the channel is from malicious parties. Without authentication, the ideas of privacy and integrity are moot.

In this thesis we consider authentication between two nodes, say Alice and Bob, who share a secret. This is the assumption used by message authentication codes (MACs), where the secret is used to generate the MAC from the message. Alice and Bob want to authenticate in the presence of an active adversary, Eve, who tries to remove, spoof, impersonate, or otherwise disrupt the authentication. A fundamental weakness in MACs lies in the secrecy of the key: repeated use of the same key reveals key information to Eve, who then can use it to disrupt the authentication. Unfortunately, this leakage of key information is unavoidable in authentication [1].

Modern authentication techniques use cryptographic operations (e.g., block ciphers, digital signatures, keyed hashes) that deter the adversary from defeating the system since doing so would require tremendous amounts of computation and

memory. However, when the abilities of the adversary increase, such authentication paradigms become more susceptible to defeat and therefore the keys must be replaced more often.

In this thesis we attack this problem by presenting novel methods for authentication and key replacement.

- We describe how to exploit the randomness of the physical layer to hide the authentication from adversaries. Using this technique reveals significantly less key information than traditional authentication methods and can increase the data throughput of the system.
- We describe a novel method of key replacement and generation that, unlike other methods, requires no additional message exchanges after initialization and yet generates highly random keys. As an added benefit, the method is shown to be extremely lightweight in terms of computation and memory requirements.

Before going into further detail, we consider the following motivating examples.

## 1.2 Motivating Examples

### 1.2.1 Cryptography is Vulnerable to Advances in Adversary Ability

Most authentication methods in use today rely on cryptography, whose security depends on the infeasibility of attacks. For example, many cryptographic algorithms rely on the hardness<sup>1</sup> of computing discrete roots, or more recently

---

<sup>1</sup>In cryptography, the hardness of a problem refers to the complexity of solving it.

elliptic curve discrete logarithms (e.g., RSA and Elliptic Curve Diffie Hellman[2], respectively). Though no efficient method is known for these problems on currently available hardware (existing methods are super-polynomial in the number of digits of the ciphertext), there are methods that can solve some problems in polynomial time on quantum computers [3], which fortunately are currently unavailable.

Historically, it has become easier to break cryptographic schemes as computers become more powerful<sup>2</sup> and less costly and as the attack algorithms become more efficient. RSA Laboratories has sponsored challenges to crack commonly used ciphers such as RC5 and DES to give security researchers understanding about the abilities of real-world attackers. As expected, short keys can be recovered in a short<sup>3</sup> time, while the recovery of larger keys can take exponentially longer<sup>4</sup> time.

However, as we noted before, the strength of the schemes are not guaranteed in the event of algorithmic or computational breakthroughs. We give a few examples of attacks against hash functions. In August 1998, a significant attack was made against SHA-0 which significantly decreased the computations required to defeat it [5]. In 2004 and 2005, more weaknesses were found in SHA-1, MD5, RIPEMD [6][7]. Most recently, in 2008, Adi Shamir (the S in RSA) introduced a new cryptanalytic technique termed the "cube attack" that exposes new vulner-

---

<sup>2</sup>Though Moore's law predicts the doubling of transistor density every 2 years, this does not translate into a doubling of performance. In fact, a  $\sim 45\%$  increase in transistors translates into only a  $\sim 10\text{-}20\%$  increase in performance [4].

<sup>3</sup>In 1999 DES, which uses a 56-bit key, was cracked in 22 hours, 15 minutes (<http://www.distributed.net/des/release-desiii.txt>)

<sup>4</sup>In 2002 RC5-64, which uses 64-bit symmetric key, was cracked in 1757 days (<http://www.distributed.net/pressroom/news-20020926.txt>)

abilities in a variety of block ciphers, stream siphers, and hash functions [8]. The typical response upon learning of these attacks is to either create new algorithms, use longer keys, or a combination of the two. However, these usually result in increased complexity for all parties involved.

We therefore wish to move towards methods whose strength does not rely solely upon the ability of the adversary. That is, in addition to cryptographic security we will also consider information-theoretic security by removing the computational constraints of the adversary.

### 1.2.2 Key Replacement is Not Cheap

Current paradigms of key replacement are expensive in different ways. We briefly outline two common methods.

1) Public key methods: The canonical key agreement example is the Diffie-Hellman (DH) protocol [2] in which Alice and Bob establish a secret key over an insecure channel. The protocol requires the exchange of 2 messages over the channel and some non-trivial computation (modular exponentiation) but the main weakness is its susceptibility to man-in-the-middle attacks: an adversary can act as an undetected intermediary between Alice and Bob and has full read and write access to all the messages shared between them. Thus DH also requires authentication methods, such as encrypted key exchange (EKE [9]), that increases the computational burden of Alice and Bob. Public key methods use asymmetric cryptography which are in general hundreds or even thousands of times slower than symmetric cryptography.

2) Infrastructure methods: Perhaps the simplest way for Alice and Bob to replace their keys is to ask someone else (an authority, such as Verisign<sup>5</sup>©) to do

---

<sup>5</sup>As an example of delegation of responsibility, Verisign's motto is: "When your customers

it for them. However, this only shifts the computational burden from the client to the server. The man-in-the-middle problem is circumvented by having the clients authenticate themselves to the trusted authority, possibly through the use of digitally signed certificates. However, this class of methods requires the presence of a third-party authority which may not always be available. In addition, the key authority becomes a single point of failure, which may not necessarily be a bad thing<sup>6</sup>.

Thus we see that the two main methods of key replacement require a combination of non-trivial computation, communications over the insecure channel, and the assistance of third parties. In this thesis we will explore how these requirements can be removed while retaining the ability to generate shared keys that appear highly random to the adversary.

## 1.3 Approach

### 1.3.1 Authentication Method

We outline our method for authentication at the physical layer. We then list the assumptions that we make.

---

trust you, trust Verisign”.

<sup>6</sup> *“We have in effect put all our rotten eggs in one basket, and we intend to watch this basket carefully”* - from the motion picture *The Great Escape*. The UNIX operating system takes this stance with regards to user passwords which through the years has proven to be a rather reasonable approach.

## Method

We concentrate our efforts at the physical layer of wireless systems. It is the first line of defense against the randomness of the environment (e.g., random noise, attenuation, and interference) and a variety of techniques are used to provide an error-free channel to the higher layers. Well-engineered systems are designed with a tolerance to handle faults that can occur during normal operating conditions. Such systems are built to fail, or be in *outage*, for only a small fraction of the time.

Typically, authentication is transmitted along with messages so that the receiver can verify both the integrity and authenticity of the messages. The authentication signal is generated from the message and a secret key shared by transmitter and receiver. Using the cryptographic terminology, we refer to the authentication signals as *tags*.

In order to have good security properties, the tag is constructed to appear random to all but the intended receiver - this protects the message from undetected forgery or tampering by making it difficult for the adversaries to construct valid message-authentication pairs. Typically, this "randomness" is achieved through cryptographic means such as encryption or keyed hashes.

Our idea for authentication is to utilize the randomness of the channel to help hide the authentication. To do this, we superimpose the authentication signal and the data waveforms. This is in contrast to time-multiplexing, which is the traditional route of cryptographic solutions. Because we are superimposing the tag on the message, unlike traditional MACs we extend the tag generation to allow null symbols. In our superimposed signal framework, this allows some portions of the data signal to be unaffected by the authentication, i.e, no superposition in some parts of the data.

In order to be useful, the authentication should have the following properties:

- have little impact on the data outage probability
- allow the receiver to reliably verify the transmitter's identity
- be difficult for any adversary to spoof or forge

Our method has the added benefit of having very low complexity.

### **Assumptions**

We make the following assumptions:

- Alice and Bob communicate using discrete packets of data
- The channel is possibly convolutive, with additive noise at the receiver
- Alice and Bob share a key unknown to anyone else
- Eve can overhear Alice and Bob's messages and transmit arbitrary messages to them

### **1.3.2 Key Replacement**

We outline our new method for key replacement. We then list the assumptions that we make.

#### **Method**

Our idea for key replacement is use key replacement models to restrict the possibilities for the next key given the current key. Without knowledge of the model, however, the key replacements appear to be completely random to the adversary. For each key, the models dictate possible sets of replacement keys. That is, for a

given key, model  $a$  may specify that the next key is either 1 or 2 while model  $b$  specifies that it is either 3 or 4.

Suppose that Alice and Bob agree upon the replacement model and the current key. Suppose further that Alice and Bob agree to change keys at certain times. To replace the key, Alice simply selects a key from the current replacement set. Since Bob knows the possibilities for the next key, Bob can test each key in the replacement set and acquire the key. Note that as long as the replacement sets are small, Bob's exhaustive search is feasible.

In order to be useful, the key replacement method should have the following properties:

- have reasonable storage and computational requirements
- allow the receiver to reliably acquire the correct key without significant expenditure of computation or energy
- keep the current key secret from any adversary
- keep the key replacement model secret from any adversary

Our key replacement method has the added benefit of not requiring any communication between transmitter and receiver, aside from initialization. While most traditional key replacement strategies require key negotiation messages to be passed, our method simply has the transmitter change the key and the receiver detect it.

### **Assumptions**

Our key replacement method makes the following assumptions:

- Initially, Alice and Bob share secrets: the secret key and which replacement model will be used

- Eve is able to observe the communications between Alice and Bob

## 1.4 Contributions and Roadmap of This Thesis

In Chapter 2 we cover the background and foundation that will be utilized in later chapters. We first introduce the insecure channel scenario and define the roles of each party. In that context, we then cover the fundamental ideas of authentication and communications over wireless channels. Finally, we define the metrics that we will use to evaluate the authentication method.

In Chapter 3 we introduce the physical layer authentication method over a single carrier by detailing the signal processing at the transmitter and receiver. We also describe the possible adversary strategies and discuss their effectiveness. We find that authentication can be reliably achieved at the physical layer while remaining hidden from adversaries. By exploiting the uncertainties of the channel to disguise the signal, we are able to keep the amount of leaked key information low. While the reliability of the authentication is particularly good in the high SNR regimes, we find that there is a tradeoff between stealth and security. We then extend the techniques to fast-fading channels and show that though the performance is decreased, authentication is still possible in such environments.

In Chapter 4 we extend the single carrier framework to the multiple carrier setting, and find that the richness of this scenario allows us to design the authentication in a very flexible manner. When multiple carriers are available for the authentication, we find that spreading the tags over many carriers increases the reliability of the authentication. At the same time, leaving some portions of the signal untagged forces Eve to estimate where the authentication is hidden - this further hinders her ability to collect information about the secret key. With some channel state information, the transmitter is able to allocate power between the

carriers as well as between the message and the tag. We find that the power allocation plays a major role in the performance of the authentication and analyze some simple strategies that yield drastically different results.

In Chapter 5 we describe our implementation of the authentication technique in a testbed of software-defined radios. After an introduction to our hardware setup, we detail the design of the transmitter and receiver as well as the experimental procedure. Finally, we demonstrate the performance of the technique and demonstrate satisfactory real-world performance.

In Chapter 6 we introduce the key replacement method that is based on Markov models. We give techniques that allow extremely large key spaces while having little storage and computational requirements. Further, we show that without knowledge of the replacement model, the adversary is unable to obtain significant information about future keys or the replacement model. Thus, the method is shown to have good security properties and be feasible for implementation. We conclude by giving examples of how this key replacement method may be used in a multitude of situations.

## Chapter 2

### Preliminaries

#### 2.1 Overview

In this chapter we lay the groundwork for the physical layer authentication method. We cover the following:

- We first describe the primary participants in the authentication and the assumptions that we make about them.
- We give an introduction to the authentication problem as well as a survey of the relevant literature.
- We give an introduction to wireless channels which motivates us to consider a method that takes advantage of the randomness for authentication.
- We introduce the relevant metrics which we will use in the following chapters for judging the performance of the new method

## 2.2 Cast of Characters

### 2.2.1 Scenario

We consider the scenario depicted in Figure 2.1 where four nodes share a wireless medium. We first describe the system where our authentication method is not used. Alice sends data **messages** to Bob using **untagged signals** while Carol and Eve listen. This network has no privacy, so Carol and Eve can understand what Alice is sending to Bob.

Now suppose that Alice and Bob use a physical layer authentication scheme (introduced in Chapter 3) that allows Bob to verify that the messages he receives are from Alice. In order to authenticate, Alice sends a proof of authentication, call it a **tag**<sup>1</sup>, simultaneously with each message for Bob's verification. The tag reflect knowledge of a key shared between Alice and Bob, and is used by Bob to verify message integrity and authenticity. We call the transmitted signal under this scheme the **tagged signal**.

Alice and Bob are the main participants of the authentication scheme. Carol and Eve are bystanders. Carol does not know the scheme and does not authenticate Alice's messages, while Eve knows the scheme but does not have the secret key. We say that Bob and Eve are **aware receivers** and Carol is an **unaware receiver**.

Authentication is a security mechanism and we must therefore consider the possible attacks on it. In the following we assume that Eve wishes to disrupt the authentication process by causing Bob to either reject authentic messages or accept inauthentic messages.

---

<sup>1</sup>We use the term "tag" to mean the authentication signal that is superimposed at the physical layer.

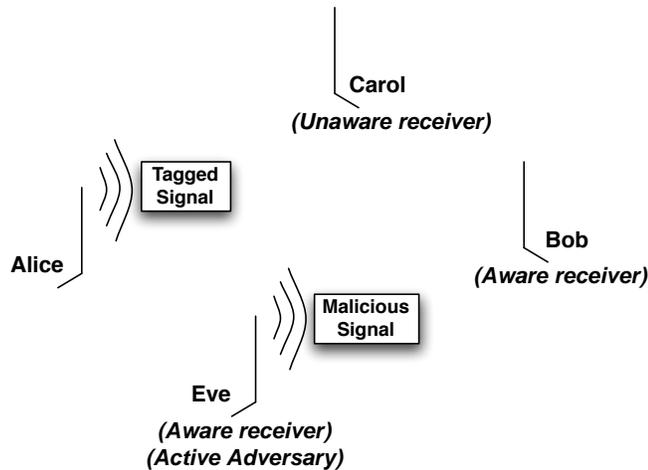


Figure 2.1: Scenario with Alice, Bob, Carol, and Eve.

## 2.2.2 Channel Assumptions

We assume that Bob, Carol, and Eve have the same type of channel (e.g., block fading), but we do not place any restrictions on the statistics. For example, Eve may have a higher average SNR than Bob.

Eve is an adversarial aware receiver who does not know the secret key. She is an active opponent and can transmit her own signals that are observable by Bob (Figure 2.1). However, it is impossible for Eve to coherently disrupt Alice’s message blocks (i.e., flip specific bits while leaving others untouched). This is a fundamental restriction at the physical layer of a mobile wireless system (Section 2.4), because any error in estimating the propagation delay, multipath, and mobility between Alice, Bob, and herself will result in a non-coherent interruption. Though Eve may try to modify specific symbols in a frame by overpowering Alice’s signal with her malicious signal, she can only corrupt the signal incoherently. Hence Eve can either transmit her own blocks or add interference to Alice’s blocks.

## 2.3 Authentication

### 2.3.1 Factors of Authentication

Authentication is the process of verifying identity. As humans, we are able to verify the identity of our friends through a variety of ways including sight and sound. When sight and sound are not available, say during text messaging, we may verify identity by obtaining satisfactory answers to a series of questions. A common question is simply ‘What is the password’.

Modern devices often authenticate in such a manner that ignores physical device characteristics such as the pulse shape, oscillator drift, etc. This is like a members-only club that requires a secret password for entry. In the absence of other safeguards, anyone who obtains the password would be granted entry into the club. Such a simple authentication scheme is totally reliant on how well the secret is kept.

Realistically, a club would not let in a stranger just because he knows the password of the day. He would be subject to additional scrutiny. A membership card may be required. His voice may be verified. His fingerprints may be scanned. In general, the factors for authentication can be divided into four categories:

- what the user *knows*
- what the user *has*
- what the user *does*
- what the user *is*

According to the U.S. Government’s National Information Assurance Glossary, ***strong authentication*** is the layered authentication approach relying on

*two or more authenticators to establish the identity of an originator or receiver of information.* Concisely, strong authentication requires multiple factors.

### 2.3.2 Common Implementations

We review examples of how the factors of authentication are commonly implemented. ‘What you know’ is proved with passwords or encryption with secret keys. ‘What you have’ is proved with smart cards, tokens, and dongles. ‘What you do’ is proved with voice recognition, gait recognition, and keystroke dynamics. ‘What you are’ is proved with fingerprint recognition, retinal scans and DNA samples.

The paradigms of ‘what you do’ and ‘what you are’ exist primarily for user-to-machine authentication; the mechanisms for verifying these paradigms for machine-to-machine authentication are not sufficiently explored. By introducing this technique at the physical layer, the way is paved for strong, multilayer authentication.

In this thesis we consider message authentication codes (MACs, or commonly *tags*), which are used to simultaneously verify message integrity and authenticity. Suppose that Alice wishes to transmit a message  $\mathbf{S}$  to Bob, and that they both share a secret key  $k$ . The tag associated with the message is

$$\mathbf{T} = g(\mathbf{S}, k) \tag{2.1}$$

where  $g(\cdot)$  is the tag generation function whose structure depends on the type of MAC<sup>2</sup>. Alice then transmits both  $\mathbf{S}$  and  $\mathbf{T}$  to Bob. Since Bob also has the secret

---

<sup>2</sup>There are many flavors of MAC that are based on different implementations: keyed-hashes (HMAC), universal hashing (UMAC), and block-ciphers (CMAC) are a few prominent examples.

key, he is able to regenerate the tag after receiving  $\mathbf{S}$ . He then compares his regenerated tag with  $\mathbf{T}$  and accepts the message as untampered and from Bob if they match exactly. Otherwise, he rejects the message. We discuss the security of such schemes in light of the possible attacks.

### 2.3.3 Attacks

To defeat authentication schemes that use MACs, Eve must be able to cause Bob to

1. reject authentic messages, *or*
2. accept inauthentic messages

with non-negligible probability.

The following are common attacks to this type of authentication system.

#### Jamming Attacks

Eve can try to distort the message or tag so that they will not be verified by Bob. She can do this by transmitting a jamming signal while Alice is transmitting to Bob. This situation may be viewed as a degradation in SNR and hence may be combatted through increased error coding or conventional physical layer methods of co-channel interference rejection. However, this attack not only destroys the authentication, but the message as well. Since authentication is only useful when the message is received correctly, the system functions as expected and the channel has other problems to deal with first before it should be concerned about authentication.

## Replay Attacks

In an effort to have Bob accept erroneous messages that do not originate from Alice, Eve can simply replay messages and tags that Alice had transmitted in the past. This is the replay attack. Eve may want to do this when Alice instructs Bob to do something favorable, e.g., ‘Pay Eve 100 dollars’. This can be combatted by assuming that Alice never transmits the same frame twice. For example, each frame may contain a timestamp or nonce<sup>3</sup>.

## Substitution Attacks

Rather than simply replaying captured message and tag pairs, Eve may send her own message block with a captured tag to Bob. The success of this attack depends on the *preimage resistance* of the function  $g(\cdot)$ . The function  $g(\cdot)$  is preimage resistant when given  $T$ , it is hard to find  $\tilde{\mathbf{S}} \neq \mathbf{S}$  such that  $g(\tilde{\mathbf{S}}, k) = g(\mathbf{S}, k) = T$ .

Since Eve does not have the key, she is unable to verify that the captured tag will match with the tag that Bob will generate given the substituted message. Hence when  $g(\cdot)$  is preimage resistant, the substitution attack is successful with very low probability.

## Impersonation Attacks

Eve may be even more aggressive and replace not only the message, but generates her own messages and tags and sends them to Bob. In this way, she attempts to impersonate Alice. Without the secret key, the probability that Eve’s message will be authenticated depends on Bob authentication test and its power.

However, when Eve gains information about the key, she is able to do better.

---

<sup>3</sup>A *nonce* is a number that is used only once. They are often random or pseudo-random numbers that are used to randomize otherwise identical input.

Note that in the worst case, Eve knows the entire key and can transmit arbitrary messages to Bob which he will accept. At this point, the system is compromised and Alice and Bob must replace the key. It is therefore of the utmost importance to keep the key secret from Eve.

### 2.3.4 Properties of the Authentication Tag

The authentication tag provides a unique identifier associated with the message that only the intended receiver can verify. To all others, the tag appears to be random so that it is difficult to predict and hence difficult to forge or spoof.

Equation (2.1) states the relationship between message, key, and tag. As Section 2.3.3 reveals, the tag generation function  $g(\cdot)$  must satisfy special properties in order for the authentication to be resistant to attack. To summarize, the tag must satisfy the following:

- Preimage resistant: given a tag it is computationally infeasible to find another message that yields the same tag
- Second preimage resistant: given a message  $x$ , it is computationally infeasible to find another message  $y$  such that  $g(x, k) = g(y, k)$ .
- Collision resistant: it is computationally infeasible to find two distinct messages  $x \neq y$  such that  $g(x, k) = g(y, k)$ .

Note that collision resistance implies second preimage resistance.

## 2.4 Wireless Channels

### 2.4.1 Channel Models

In a single-antenna system, the transmitted signals pass through a convolutive channel and are corrupted by additive noise. The receiver observes the signal:

$$y(t) = \int_0^{\infty} h(t, \tau)x(t - \tau)d\tau + \eta(t) \quad (2.2)$$

where  $x(\cdot)$  is the transmitted signal,  $h(t, \cdot)$  is the channel response at time  $t$ ,  $\tau$  is the delay parameter for the multipath components, and  $\eta(\cdot)$  is the additive noise. The channel is described by a time-variant impulse response

$$h(t, \tau) = \alpha(t, \tau)e^{-j\theta(t, \tau)} \quad (2.3)$$

where  $\alpha(t, \tau)$  is the gain and  $\theta(t, \tau)$  is the phase shift of the channel. When there are many scatterers and paths, the Central Limit Theorem is often applied to approximate the sum contribution of all paths as a Gaussian variable in  $\alpha(\cdot)$ . Also, since small changes perturbs the phase significantly, the phase is commonly modeled by a uniform  $[0, 2\pi]$  random variable. In rich multipath environments  $h(t, \tau)$  can be modeled as a zero-mean complex Gaussian process. In this model the gain has a Rayleigh distributed envelope, and is known as the Rayleigh fading channel. When a line-of-sight exists, the channel is no longer zero-mean and the envelope is modeled as a Ricean random variable instead. This model is known as the Ricean fading channel. The availability of a Rayleigh or Ricean fading is dependent on the richness of the scattering environment, and hence these assumptions are not always applicable. There are many other models that characterize the channel by autocorrelation, geometry, or other methods. A widely used model is the Jakes' model (also known as Clarke's model), which describes the situation when a mobile receiver is surrounded by a uniform (with respect to angle)

scattering environment. We are concerned primarily with the Rayleigh fading channel in this thesis.

With a symbol time  $T$ , sampled the channel at rate  $1/T$  yields the the received symbol

$$y_k = \sum_i h(k, i)x_{k-i} + n_k \quad (2.4)$$

This is only one of the many ways to discretize the observation. In general the continuous signal can be represented by its projection onto basis functions. For example, we may use the Karhunen-Loeve [10] or various wavelet expansions [11] to acquire fine information about the signal. Therefore in this thesis we view the discrete signal to be in general a result of a basis decomposition.

Two important channel parameters are coherence time and coherence bandwidth. When two parts of the signal are separated by more than the coherence time (or bandwidth), they experience independent channels. When the coherence time is large, the channel is said to be slow-fading and may be assumed constant over many symbols. Similarly, when the coherence time is small, the channel is fast-fading and may be assumed constant for only a few symbols. When the signal bandwidth is smaller than the channel coherence bandwidth, the signal is narrowband and experiences flat fades, i.e., constant fade across all frequencies. In this case, the channel at time  $k$  is  $h(k, i) = h(k)\delta_i$ , since the effect of the channel is localized in time. If greater, the signal is wideband and experiences frequency-selective fades so  $h(k, i)$  is a vector for each  $k$ . For example, an OFDM signal occupies a bandwidth that experiences frequency selective fades, but each carrier can be narrow enough so the fading is approximately flat over its band.

The signals are conventionally viewed in blocks of length  $L$ :  $\mathbf{x} = \{x_1, \dots, x_L\}$ . The observation can be written in matrix notation

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{N} \quad (2.5)$$

In the case of frequency flat fades,  $\mathbf{H}$  is a diagonal matrix with elements  $h(k, i) = h_k \delta(k - i)$ , i.e.

$$\mathbf{H} = \begin{bmatrix} h_1 & & 0 \\ & \ddots & \\ 0 & & h_L \end{bmatrix} \quad (2.6)$$

When the channel is time-invariant over the block,  $h_1 = h_2 = \dots = h_L$ .

For frequency-selective fades, we have in general a matrix populated with non-zero entries

$$\mathbf{H} = \begin{bmatrix} h_{11} & \dots & h_{1L} \\ \vdots & \vdots & \vdots \\ h_{L1} & \dots & h_{LL} \end{bmatrix} \quad (2.7)$$

When the channel is time-invariant over the block,  $\mathbf{H}$  is Toeplitz.

The time variation of a channel may be introduced using simple mechanisms. The simplest is to assume that the channel is time-invariant over the symbol block, but each channel realization is different from block to block. At one extreme, the channel may be constant over an infinite number of symbols (slow fading) or only a single symbol (fast fading). The channel realizations from block to block depends on the coherence time; they may be modeled as independent or correlated. A simple class of models are the Finite State Markov Channels (FSMC) which are typically used to model channel amplitudes. A FSMC model for Rayleigh Fading introduced by Wang [12] discretizes the SNR  $\gamma$  into regions such that the SNR stays within a region during a time interval  $T$ . For the next time interval, the SNR can remain in the region or transition to adjacent regions. The transition probabilities are dependent on the level crossing rates at the boundaries. One particular case are the autoregressive (AR- $n$ ) models, where the current channel is correlated with the  $n$  immediately previous channel instances.

In this thesis we consider the independent block fading model for slow-fading case and the AR-1 channel for the fast-fading case.

## 2.4.2 Uncertainties

Aside from the actual message content, the receiver experiences uncertainty in many places. Recall that the received signal is modeled by a convolutive channel and additive noise (equation (2.2)). Because of imperfect channel estimation and the additive noise, the receiver cannot be certain of what signal was transmitted. Therefore we propose hiding the authentication by superimposing it on the message signal, where its presence is obscured by the channel estimate and noise. We now elaborate on these sources.

### Channel Estimation

In order to estimate the channel, the receiver needs to know something about the transmitted signal and how the channel distorted it. In blind estimation, the actual transmitted signal is not known but statistics or other characteristics of it are known. For example, knowing that constant modulus signals are being transmitted is very helpful for channel estimation [13].

We use pilot symbols to estimate the channel. Pilot symbols are known data that are used to help the receiver compensate for channel effects. The power, position, and design of the pilot symbols affect the performance of system through the quality of the channel estimate and the complexity of the receiver [14].

Traditionally, pilots are placed periodically in the signal and interpolated to provide a reasonable estimate for the channel. For example, in a slowly-fading channel, pilots may be placed in the center of a block and the estimated channel used for symbols before and after the pilots. Optimal TDM schemes that

maximize mutual information require a substantial overhead and may not even provide good channel estimates [15]. In the low SNR regime with short coherence time and hence short block lengths, much of the block is spent estimating the channel. However, with high SNR and long coherence times, TDM schemes perform very well.

There are many approaches to channel estimation, the common ones are minimum mean square (MMSE) and maximum likelihood (ML). Let the subscript  $(\cdot)_p$  denote the vector of pilot symbols. For example,  $\mathbf{x}_p$  are the transmitted pilot symbols while  $\mathbf{y}_p$  are the corresponding observations through the noisy channel. A linear channel estimate is a linear function of the observation:

$$\hat{\mathbf{h}} = \mathbf{W}\mathbf{y}_p \quad (2.8)$$

where  $\mathbf{y}_p$  are the observations of the pilot symbols  $\mathbf{s}_p$ . The linear full-rank estimator that minimizes the mean square error is the Wiener filter:

$$\hat{\mathbf{h}}_{\text{LMMSE}} = \arg \min_{\mathbf{x}} E\|\mathbf{x} - \mathbf{h}\|^2 \quad (2.9)$$

For zero mean channels, the solution is well known to be

$$\hat{\mathbf{h}}_{\text{LMMSE}} = (\mathbf{R}_n \mathbf{I} + \mathbf{R}_h \mathbf{x}_p^H \mathbf{x}_p)^{-1} \mathbf{R}_h \mathbf{x}_p^H \mathbf{y}_p \quad (2.10)$$

For a channel with Gaussian statistics, the Kalman filter is a LMMSE in steady state.

In AWGN, the maximum likelihood estimator is

$$\hat{\mathbf{h}}_{\text{ML}} = \arg \min_{\mathbf{x}} \|\mathbf{y}_p - \mathbf{x}_p \mathbf{x}\|^2 = (\mathbf{x}_p^H \mathbf{x}_p)^{-1} \mathbf{x}_p^H \mathbf{y}_p \quad (2.11)$$

The ML estimator is the zero-forcing solution in that the error at the pilot symbol positions is zero. However, one of its weaknesses is that it can significantly enhance noise over the message symbols.

Without perfect channel state information (CSI) at the receiver, no channel estimate can be free from error. The uncertainty of the estimate is usually measured by mean-squared error and is a function of 1) how well the channel model fits the real scenario, 2) the strength of the noise and interference (SINR), and 3) the pilot symbols and the estimation scheme.

## Noise Models

The most common assumption about the additive noise  $\eta(t)$  is that it is a white Gaussian process. This process is a purely mathematical construct since it is characterized by a constant power density of  $N_0/2$ , and hence infinite power. The noise power at the receiver is limited by the receiver bandwidth. This additive white Gaussian noise (AWGN) assumption is commonly used to describe thermal noise at the receiver or a large aggregate of interferers by invoking the Central Limit Theorem.

However, the AWGN assumption is inadequate to model impulsive noise. Impulsive noise comes from many sources, both natural (e.g. lightning) and man-made (e.g. other transmitters, microwave ovens). The basic model of this noise is as a train of pulses characterized by random amplitude, duration, and interarrival time as described by the Amplitude Probability Distribution (APD), Pulse Duration Distribution (PDD), and the Pulse Spacing Distribution (PSD). The most basic way of differentiating between Gaussian and impulsive noise is to classify noise above a certain threshold as an impulse. The pulse duration is defined as the time between the positive and subsequent negative crossing. Thus a related characterization of impulse noise is the Average Crossing Rate (ACR), or the rate of positive crossings.

Assuming that the impulses are independent and can occur at any time, the

pulse occurrences are fundamentally Poisson. The pulse duration and amplitude may either be modeled by random variables or determined empirically. For example, the pulse amplitude has been approximated by a power Rayleigh pdf and the pulse duration by a Gamma pdf. There have been numerous measurements and techniques [16][17] as well as statistical models and performance analysis [18][19].

Since the noise is inherently random, it is characterized by its probability distribution. Therefore in order for the authentication to be hidden by the noise, it should follow the same distribution; the authentication should be Gaussian when the additive noise is Gaussian, and impulsive when the noise is impulsive. This is a simple matter when the noise distribution is known; however, often-times the distribution is a rough approximation and there is a discrepancy. The ability of the receiver to discriminate between noise and signal (authentication) is discussed in Section 2.5.2.

### **2.4.3 Capacity vs. Rate**

We have broken from the traditional TDM placement of authentication tags and decided to superimpose them instead (see Figure 2.2 for comparison). The benefits of simultaneously transmitting message and tag are manifold.

- Message symbols are constantly transmitted; at no time is message transmission halted to transmit tags. This is typical of non-authenticated communications where messages are queued up and transmitted as soon as possible.
- Careful superposition of the tag can be transparent to the receivers' implementation. That is, the operation of the receiver does not change in the presence or absence of the authentication tag.

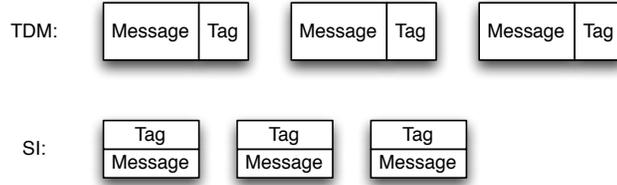


Figure 2.2: Time division multiplexed (TDM) versus superimposed (SI) tags

- Capacity that is available but not normally not utilized can be exploited in higher SNR.

The fundamental limitation on rate is the capacity of the channel. As the SNR of the channel increases, the capacity increases as well. However, when a fixed modulation scheme is imposed on the channel, the rate saturates at a finite limit. Thus as the SNR increases, the unexploited capacity (or capacity headroom) grows. Our proposed scheme exploits this headroom. To maintain robustness in fading environments (where SNR changes with time) a modulation scheme is chosen where the rate is flat within the expected range of SNR variation. By superimposing additional symbols, the achievable rate is higher when the SNR is suitably high.

For a simple example refer to Figure 2.3, which shows the rate of BPSK with equiprobable values  $\rho_s \in \{\pm 1\}$  in an additive white Gaussian noise (AWGN) channel. The vertical region between the rate and capacity curves is the headroom. Now when we superimpose an additional independent bit  $\rho_t$ , we use the voltages  $\{\pm\rho_s \pm \rho_t\}$  to signal 2 bits at a time. We scale  $0 \leq \rho_s, \rho_t \leq 1$  to satisfy  $\frac{1}{2}(\rho_s^2 + \rho_t^2) = 1$  for energy fairness. We see that  $\rho_s^2$  percent of the original energy is used to signal the first bit while  $\rho_t^2$  percent is used to signal the second. As described before, the rates of this new scheme exceed that of BPSK, and the improvement grows as the SNR or  $\rho_t^2$  increases.

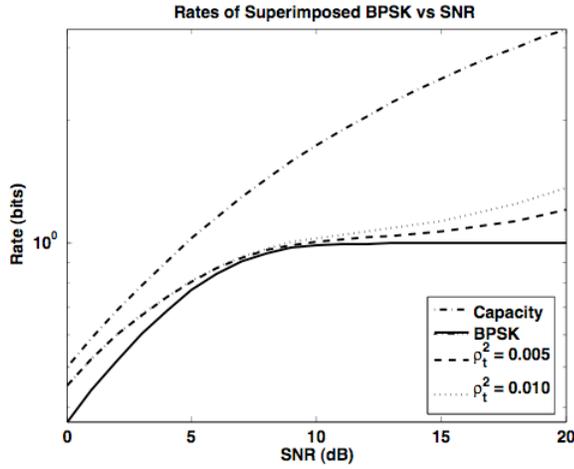


Figure 2.3: Rate of superimposed BPSK

## 2.5 Evaluation Metrics

In this section we consider how we evaluate the new authentication method. The proposed method works outside traditional paradigms and so requires additional metrics to measure performance:

- **Stealth:** the authentication is difficult to detect for everyone but the intended receiver. In particular, it should not significantly impact the performance of the existing system.
- **Robustness:** the authentication can be reliably detected by the intended receiver
- **Security:** the authentication is difficult to forge or spoof. In particular, the adversary should not be able to gain significant amounts of information about the secret key.

The metrics will rely on hypothesis testing, so to preface the discussion of the metrics we introduce the fundamentals of hypothesis testing in the following

section.

### 2.5.1 Hypothesis Testing

One of the most fundamental tests in hypothesis testing stems from the Neyman-Pearson lemma, which yields the formulation of threshold tests.

**Neyman-Pearson lemma:** Let  $X_1, X_2, \dots, X_n$  be drawn i.i.d. according to a probability mass function  $Q$ . Consider the decision problem corresponding to hypotheses  $Q = P_1$  versus  $Q = P_2$ . For a threshold  $T \geq 0$ , define a region

$$A_n(T) = \left\{ \frac{P_1(x_1, x_2, \dots, x_n)}{P_2(x_1, x_2, \dots, x_n)} > T \right\} \quad (2.12)$$

Let

$$\alpha^* = P_1^n(A_n^c(T)), \beta^* = P_2^n(A_n(T)) \quad (2.13)$$

be the corresponding probabilities of error corresponding to decision region  $A_n$ . Here  $\alpha^*$  is the probability of observing a sequence in  $A_n^c(T)$  (i.e., outside of region  $A_n(T)$ ), when the sequence is drawn from distribution  $P_1$ . Similarly,  $\beta^*$  is the probability of observing a sequence in  $A_n(T)$  when the sequence is drawn from distribution  $P_2$ .  $\alpha^*$  is also known as a type I error while  $\beta^*$  is a type II error. Let  $B_n$  be any other decision region with associated probabilities of error  $\alpha$  and  $\beta$ . If  $\alpha \leq \alpha^*$  then  $\beta \geq \beta^*$ .

The likelihood ratio test is equivalent to

$$D(P_{X^n}||P_2) - D(P_{X^n}||P_1) > \frac{1}{n} \log T \quad (2.14)$$

where  $D(f||g)$  is the Kullback Leibler distance between probability densities  $f$  and  $g$  defined as

$$D(f||g) = \int f \log \frac{f}{g} \quad (2.15)$$

The following lemma upper bounds the error exponent in hypothesis testing by the distance between the distributions of each hypotheses. The basic intuition is that the further the hypotheses are from each other, the easier it is to distinguish between them.

Stein's Lemma: Let  $X_1, X_2, \dots, X_n$  be i.i.d. according to a probability density function  $Q$ . Consider the hypothesis test between two alternatives,  $Q = P_1$  and  $Q = P_2$ , where  $D(P_1||P_2) < \infty$ . Let  $A_n \subseteq \mathcal{X}^n$  be an acceptance region for hypothesis 1. Let the probabilities of error be

$$\alpha^* = P_1^n(A_n^c(T)), \beta^* = P_2^n(A_n(T)) \quad (2.16)$$

and for  $0 < \epsilon < \frac{1}{2}$ , define

$$\beta_n^\epsilon = \min_{\substack{A_n \subseteq \mathcal{X}^n \\ \alpha_n < \epsilon}} \beta_n \quad (2.17)$$

Then

$$\lim_{\epsilon \rightarrow 0} \lim_{n \rightarrow \infty} \frac{1}{n} \log \beta_n^\epsilon = -D(P_1||P_2) \quad (2.18)$$

## 2.5.2 Stealth

We say the authentication scheme is stealthy if it is difficult to detect. There are actually two aspects to stealth.

1. the *presence* of the tag should not be easily detectable by adversaries
2. the *impact* of the tag should not be felt by the unaware receivers

### Authentication Presence

The first aspect is measured by the ability of the adversary to detect the authentication. Recall that the adversaries do not have the secret key. Using Stein's

lemma, we fix the false alarm probability  $\alpha$  and investigate how the missed detection probability  $\beta$  is affected by the authentication.

The receiver decides between the hypotheses

$$H_0 : \quad \text{Observation does not contain any authentication} \quad (2.19)$$

$$H_1 : \quad \text{Observation contains authentication} \quad (2.20)$$

Let  $P_i$  be the density of the observations under  $H_i$  and let  $A_n$  be the acceptance region for hypothesis 1 such that the false alarm probability is  $\alpha_n < \epsilon$ .

Applying Stein's lemma, by fixing the false alarm probability  $\alpha_n$ , the missed detection probability  $\beta_n$  decreases with an error exponent  $D(P_0||P_1)$ , the KL distance between the two possible distributions. Thus if the two distributions have large KL distance, a few observations are sufficient to make a good decision, while if the distance is small, many observations are necessary. Hence in order for the tag presence to be stealthy, the probability density of the tagged signal should be close in KL distance to the untagged signal's density.

In reality, Eve does not calculate the KL distance in order to determine the presence of authentication. She can quantify the discrepancy between her observation and the baseline distribution by using goodness-of-fit tests [20]. These tests decide whether or not the data follows the given distribution at a significance level  $\alpha$ . The *Kolmogorov-Smirnov test* is one example, and its statistic is

$$D = \max_{1 \leq i \leq N} \left( F(Y_i) - \frac{i-1}{N}, \frac{i}{N} - F(Y_i) \right) \quad (2.21)$$

which is simply the greatest discrepancy between the observed and expected cumulative frequencies. The null hypothesis (data does not follow distribution) is accepted if the statistic exceeds a critical value obtained through table lookup. Note that the critical values do not depend on the particular distribution being tested. The outcome of the K-S test is only valid when the distribution is

fully specified, i.e., the parameters of the distribution cannot be estimated from the data. When the parameters are not known, the *Anderson-Darling test* may be applied, though the critical values are available only for a few distributions. For an assumption of Gaussian noise, the *Lilliefors test* may be used when the expected value and variance are estimated from the samples.

### Authentication Impact

Even if the authentication tag is difficult to detect by the adversary, it may not necessarily be stealthy. To see this, consider the following example. Suppose that a tag symbol  $t$  is added to a BPSK message  $s$  and transmitted through an AWGN channel with SNR  $\gamma$ :

$$y = s + t + w \quad (2.22)$$

$$s \in \{\pm 1\} \quad (2.23)$$

where  $w$  is a zero-mean Gaussian random variable with variance  $\sigma_w^2 = 1/\gamma$ .

Assume that the receiver knows that the noise is Gaussian, but does not know the power a priori. However, he is able to estimate the noise power from his observations. If the tag is a zero-mean Gaussian random variable with variance  $\sigma_t^2$ , then the estimated noise power  $\hat{\sigma}_w^2$  is zero-mean Gaussian with variance  $\sigma_t^2 + \sigma_w^2$ . Thus the unaware receiver treats the tag as noise and is unable to detect its presence, regardless of its power.

However, as the tag power increases the message throughput drops to zero. Thus the stealth requirement also restrains the authentication from having a significant impact on the message recovery. This is dependent on the choice of modulation and authentication.

The basic metrics for this aspect of stealth are bit error rate and frame outage probability. We constrain the impact of the authentication by limiting the

allowable deviation from the baseline BER and outage probability.

### 2.5.3 Robustness

We say that the authentication scheme is robust if it can operate in a wide range of environments. That is, the intended receiver (Bob) should be able to verify the authentication with a given confidence despite fluctuating channel and noise conditions.

Suppose that Bob receives the message correctly and wants to authenticate its source. Phrased as a hypothesis test, Bob needs to determine if the correct authentication tag is present in his observation or if it contains only random noise or an incorrect tag. He decides between the hypotheses

$$H_0 : \quad \text{Observation does not contain the correct tag} \quad (2.24)$$

$$H_1 : \quad \text{Observation contains the correct tag} \quad (2.25)$$

Recall that since Bob knows the secret key, he is able to generate the authentication tag corresponding to the message. Fixing the false alarm probability  $\alpha$ , the missed detection probability  $\beta$  decreases with the error exponent  $D(P_0||P_1)$  where  $P_i$  is the density of the observations under  $H_i$ .

Actual implementation of the tag detection depends on the modulation of the tag, the estimation of the channel, and the estimation of the message. There is a fundamental tradeoff between stealth and effectiveness: the easier it is for Bob to determine the tag, the easier it is for Eve to detect its presence. However, the tradeoff may not always be in favor of Eve. Hence the notion of security is required and we will discuss it in the following section.

The intended receiver has knowledge of the secret key, and when we have stealth, the message is recovered with little increase error. In practical systems,

the error correction codes will mitigate the impact of the authentication on the data outage. As long as the message is recovered without error, the intended receiver can generate the correct authentication tag using equation (2.1).

In AWGN, Bob can use a matched filter to form a test statistic for use in a threshold test (Section 2.5.1). By fixing the false alarm probability  $\alpha$ , we take the power of the test ( $1-\beta$ , i.e., the probability of detecting the authentication when it is present) as the indication of robustness.

## 2.5.4 Security

We say that an authentication scheme is secure when Eve cannot defeat the system. More precisely, Eve should be able to successfully attack the system (Section 2.3.3) with only a negligible probability.

Recall that authentication is a hypothesis test (equations (2.24) and (2.25)). In this context, Eve's attacks are successful if Bob rejects authentic messages (forcing a type II error) or accepts incorrect messages (forcing a type I error).

We reiterate our problem setup. Alice transmits public messages  $S \in \mathcal{S}$  to Bob. In order to authenticate the public messages, she also sends authentication tags  $T \in \mathcal{T}$ , one tag for each message. Each tag is generated with a secret key  $k$  and may depend on the current and prior public messages or tags:  $T_i = f(k, S_1, \dots, S_i, T_1, \dots, T_{i-1})$ . In the following we assume that the messages and tags are transmitted through a noiseless channel.

Consider an arbitrary random variable  $U \in \mathcal{U}$  whose distribution is  $P$  under  $H_0$  and  $Q$  under  $H_1$ . Suppose that  $H_0$  is true. Then the test will decide  $H_0$  with probability  $1 - \alpha$  and  $H_1$  with probability  $\alpha$ . Similarly, when  $H_1$  is true, the test will decide  $H_0$  with probability  $\beta$  and  $H_1$  with probability  $1 - \beta$ . The binary

discrimination between these two distributions is

$$d(\alpha, \beta) = \alpha \ln \frac{\alpha}{1 - \beta} + (1 - \alpha) \ln \frac{1 - \alpha}{\beta} \quad (2.26)$$

*Theorem [21, Theorem 4.3.3]:* Discrimination is increased by a nontrivial refinement of a set of measurement outcomes.

From this theorem, a well-known result follows:  $D(P||Q) \geq d(\alpha, \beta)$  since the distribution of the hypothesis test outcome may be viewed as a refinement of the observation probability distribution. The utility of this will soon become clear as we discuss impersonation and substitution attacks.

First we describe how Alice and Eve's generate their messages. Alice's transmission  $(T_i, K)$  is generated by the probability distribution

$$P_{T_i, K | T_1=t_1, \dots, T_{i-1}=t_{i-1}} \quad (2.27)$$

However, Eve does not have any knowledge of the secret key and so the pair is generated by a product of marginal distributions

$$Q_{T_i | T_1=t_1, \dots, T_{i-1}=t_{i-1}} \cdot P_{K | T_1=t_1, \dots, T_{i-1}=t_{i-1}} \quad (2.28)$$

where  $Q$  is the probability distribution that dictates Eve's attack strategy and  $P$  is the strategy that Alice uses. Note that Eve has access to Alice's strategy, but not the secret key. This states that Eve chooses the tag and key based on what she has seen so far  $(T_1, \dots, T_{i-1})$ .

### Impersonation Attack

*Theorem [1, Theorem 3]:* For every authentication scheme and for every particular values of  $t_1, \dots, t_{i-1}$ , we have

$$d(\alpha, P_{I,i}(t_1, \dots, t_{i-1})) \leq I(T_i; K | T_1 = t_1, \dots, T_{i-1} = t_{i-1}) \quad (2.29)$$

Moreover,

$$d(\alpha, P_{I,i}) \leq I(T_i; K | T_1, \dots, T_{i-1}) \quad (2.30)$$

In particular, for  $\alpha = 0$  we have

$$P_{I,i}(t_1, \dots, t_{i-1}) \geq 2^{-I(T_i; K | T_1=t_1, \dots, T_{i-1}=t_{i-1})} \quad (2.31)$$

and

$$P_{I,i} \geq 2^{-I(T_i; K | T_1, \dots, T_{i-1})} \quad (2.32)$$

One admissible strategy is for Eve to choose her authentication tag based on  $Q_{T_i | T_1=t_1, \dots, T_{i-1}=t_{i-1}} = P_{T_i | T_1=t_1, \dots, T_{i-1}=t_{i-1}}$ . Then the above theorem follows naturally since for random variables  $X, Y$  with joint density  $P_{XY}$  and marginals  $P_X, P_Y$ , we have  $D(P_{XY} || P_X \cdot P_Y) = I(X; Y)$ . The theorem gives the result that the more key information present in the authentication tags, the lower the probability of a successful impersonation attack. The intuition is that when tags contain little key information, it is easy for Eve to randomly guess a valid tag.

### Substitution Attack

A particular substitution attack attempts to guess the secret key, and thus gain the ability to forge authentication tags at will. For an arbitrary random variable  $U$ , note that  $H(U) \geq -\log_2(\max_u P_U(u))$  and hence  $\max_u P_U(u) \geq 2^{-H(U)}$ . This may be interpreted as the probability of guessing a realization of  $U$  correctly given that its probability distribution is known. Similarly, when side information is given in the form of random variable  $V$ , we have  $\max_u P_{U|V=v} \geq 2^{-H(U|V=v)}$ .

*Theorem [1, Theorem 6]:* For all  $t_1, \dots, t_i$  we have

$$P_{S,i}(t_1, \dots, t_{i-1}) \geq 2^{-H(K | T_1=t_1, \dots, T_{i-1}=t_{i-1})} \quad (2.33)$$

Moreover,

$$P_{S,i} \geq 2^{-H(K | T_1, \dots, T_{i-1})} \quad (2.34)$$

The theorem gives the result that the higher the uncertainty of the secret key after a sequence of authentication tags, the lower the probability of a successful substitution attack.

### Key Equivocation

We see that the strength of the authentication is dependent on how well the secret key is guarded. For if the secret key was known to the adversary, she could transmit arbitrary messages to Bob that will be accepted as genuine. Therefore we consider the *key equivocation* of the Eve: the adversary's uncertainty about the key. For example, if Eve has a key equivocation of 64 bits about a 64 bit key, then she has no information about it, whereas if she has 0 equivocation then she knows the entire key without any uncertainty.

We consider the key equivocation for two cases: when the tag is known without error and when it is corrupted.

#### Known Tag

When the tag is known to the receiver, the key equivocation is

$$H(K|S, T) = \sum_{s,t} p(S = s, T = t) H(K|S = s, T = t) \quad (2.35)$$

Expanding the last term with the shortened notation  $p(k|s, t) = p(K = k|S = s, T = t)$ , we have

$$H(K|S = s, T = t) = \sum_k p(k|s, t) \log_2 \left( \frac{1}{p(k|s, t)} \right) \quad (2.36)$$

we see that the key equivocation is positive only when  $p(k|s, t) < 1$  for at least one pair of  $S, T$ . That is, multiple keys map the same message to the same tag. This is guaranteed by the pigeonhole principle when either  $|K| > |T|$  or  $|K| > |S|$  (see Figure 2.4). Usually in cryptographic applications the key is much shorter than the message or tag so that positive key equivocation is not guaranteed.

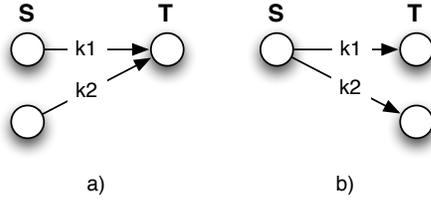


Figure 2.4: Pigeonhole principle. a) If another key is added, the number of keys will exceed the number of messages, there will be multiple keys that map at least 1 pair of (message,tag). b) A similar situation arises when the number of keys exceeds the number of tags.

Let us consider the significance of having zero key equivocation given the message and tag (i.e.,  $H(K|S, T) = 0$ ). This allows us to say that given  $S$ , knowing  $T$  is equivalent to knowing  $K$  and vice versa. Here we assume that the mapping from  $\mathcal{S} \times \mathcal{K} \rightarrow \mathcal{T}$  is deterministic so that  $H(T|S, K) = 0$ . That is,

$$\begin{aligned}
 I(K; T|S) &= H(K, T|S) - H(K|S, T) \\
 &\quad - H(T|S, K)
 \end{aligned} \tag{2.37}$$

$$= H(K, T|S) \tag{2.38}$$

Thus in terms of information theory, knowing the tag is just as good as knowing the key. Theoretically therefore, it is very serious for the adversary to know the tag. In cryptographic applications this is exactly the case since all parties have noiseless access to everything that is transmitted. However, cryptological security relies on the infeasibility of computational attack, which may be improved through increasing key length or increasing the complexity of the solving the underlying cryptographic primitive<sup>4</sup>.

---

<sup>4</sup>Cryptographic primitives can be thought of as a cryptographer's toolbox; primitives are low-level, well-established algorithms that are frequently used in security systems. Some examples

We are interested in what happens as the constraints of the adversary are removed, and therefore we turn our attention to cases where the tag is not known exactly.

### **Estimated Tag**

When the tag is estimated by the receiver, it may be recovered with error. Since Eve estimates each tag symbol with some non-zero error, the size of her search space increases with the tag symbol equivocation. A straightforward solution is to compute the tags corresponding to each possible key (there are  $2^K$ ), then select the key that generates the signal most similar to the residual. This is the brute force method.

Actually, Eve does not have to check all  $2^K$  keys before she finds the correct key. The birthday paradox<sup>5</sup> states that on average, she only has to check on the order of  $2^{K/2}$  keys before she finds the correct key. This assumes that the tags are uniformly distributed; if not, then the key can become more vulnerable to discovery [22]. However, even with the paradox, with a sufficiently large  $K$  the brute force method is impractical because Eve will run into computation and memory restraints. The remaining alternative is to attempt inversion of the tag generating function  $g(\cdot)$  from equation (2.1).

When the image of  $g(\cdot)$  is observed with sufficient length and without noise, Eve may be able to recover the key in reasonable time. This would be a real concern in the higher layers. However, we use  $g(\cdot)$  in the physical layer where there is always some uncertainty about the tag. The adversary has no choice but to spread its key recovery efforts among the probable tags. For binary tag

---

include one-way hash functions and various block ciphers.

<sup>5</sup>The birthday paradox is so-named because of the party trick where in a room with 23 people, the probability where at least two people share a birthday is approximately 1/2, a surprising result to many people.

symbols, the number of possible transmitted words doubles as each tag symbol is estimated. The receiver must prune the possibilities to consider only the more probable tags, otherwise all possible tags would be considered.

The set of probable tags depends on the tag bit error probability  $p$ . When  $p$  is small, the paths that include few errors should be considered more probable, while the opposite is true when  $p$  is large. For example, suppose that the receiver estimates the binary tag sequence 000. When  $p$  is small, the most likely transmitted sequence is 000, and the second most likely sequences are  $\{001, 010, 100\}$ . The least likely transmitted sequence is 111. If we have a length- $L$  observation and choose to consider paths with  $k$  or fewer errors, we expand the search space by  $\sum_{i=0}^k \binom{L}{i}$ , which is a polynomial factor for fixed  $k$ .

Let us suppose that there is zero key equivocation when the tag is known exactly, i.e.,  $H(K|S, T) = 0$  so that knowing the tag is equivalent to knowing the secret key. The key equivocation is then equivalently

$$H(K|S, \hat{T}) = H(T|S, \hat{T}) \quad (2.39)$$

$$= \sum_{s, \hat{t}} p(S = s, \hat{T} = \hat{t}) H(T|S = s, \hat{T} = \hat{t}) \quad (2.40)$$

We know that the last term  $H(T|S = s, \hat{T} = \hat{t})$  depends on  $p(t|s, \hat{t}) = p(T = s|S = s, \hat{T} = \hat{t})$  which is

$$p(t|s, \hat{t}) = \frac{p(t|s)p(\hat{t}|s, t)}{p(\hat{t}|s)} \quad (2.41)$$

$$= \frac{p(t|s)p(\hat{t}|t)}{p(\hat{t}|s)} \quad (2.42)$$

We assume that keys are chosen uniformly so that  $p(t|s) = 1/K$  if there exists a key that maps  $s$  to  $t$  and 0 otherwise. Note that if no such key exists, then  $p(t|s) = 0$ .

To simplify the discussion, suppose that  $|S| = |T| = |K| = K$  and assume that

$p(\hat{t}|s) = p(\hat{t}) = 1/K$ . That is, the probability of observing any tag is equiprobable and independent of which message was transmitted. Then we have the following simplifications

$$p(S = s, \hat{T} = \hat{t}) = \sum_{s, \hat{t}} p(s)p(\hat{t}|s) = 1/K^2 \quad (2.43)$$

$$p(t|s, \hat{t}) = p(\hat{t}|t) \quad (2.44)$$

and therefore the key equivocation (equation (2.40)) becomes

$$H(K|S, \hat{T}) = \sum_{\hat{t}, t} p(\hat{t}|t) \log_2 \left( \frac{1}{p(\hat{t}|t)} \right) \quad (2.45)$$

Notice that the key equivocation is thus dependent on the reliability of the tag observation  $p(\hat{t}|t)$ . In the best case for security, the observed tags are equiprobable and the adversary cannot recover any key information; in the worst case the probability distribution is a delta function and the adversary recovers the key perfectly.

## 2.6 Related Work

### 2.6.1 Cryptographic and Information Theoretic Security

A system is considered *cryptographically* secure if an adversary cannot defeat the system given certain constraints on computational ability and time. A system is considered *information-theoretically* secure if the system cannot be defeated even when those constraints are removed. Information theoretic security is obviously a stronger requirement than cryptographic security.

In order to discuss security we review the concept of *perfect secrecy*. A message is passed in perfect secrecy when the adversary has equal *a priori* and *a posteriori* probabilities of knowing the content after observing the ciphertext. That is, the

observations do not help the adversary make a better estimate of the message. In our problem, we assume that the messages are public to all while the security lies in the tags. Thus we are interesting in protecting the secrecy of the tags against removal, modification, and impersonation.

There are two classical views of a secrecy system: Cryptology generally assumes noiseless observations of message and tags, while Wyner's wiretap channel [23] assumes noisy observations. Cryptology attempts to obscure the meaning of the noiseless observations, while Wyner's wiretap channel attempts to obscure the observations of the secret data. We briefly walk through the scenarios and security results of each paradigm.

## **Cryptology**

In cryptology, messages and tags are transmitted over a perfect channel so that any receiver, including the adversary, can observe them without error. The security of a cryptographic system lies in the complexity of inversion without the secret key. Such systems heavily use one-way functions that are easy to compute in one direction but difficult (in terms of computation and memory requirements) to invert. Many are based on mathematical primitives that to date do not have efficient inversion algorithms such as discrete logarithms and elliptic curves.

Along with the idea of complexity is the issue of search space. By spreading out the secret keys and tags over a large multidimensional space, the task of the adversary becomes much harder. If the space is large enough, the adversary cannot test every possible key given current (and hopefully future) limitations on computation power and memory availability.

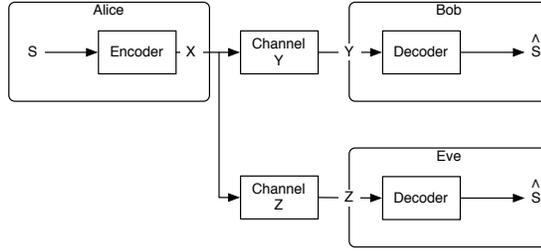


Figure 2.5: Wiretap Channel

### Wiretap Channel

A generalization [24] of Wyner’s well-known wiretap channel follows. The authentic sender encodes the symbols into the transmitted signal  $x \in \mathbb{R}^P$  (where  $P$  is the length of the message) using the stochastic encoder  $f(\mathbf{x}|\mathbf{s}, \mathbf{t}) : \mathcal{S} \times \mathcal{T} \rightarrow \mathbb{R}^T$ . The signal observed by an aware receiver is modeled by instances of the probability density  $q_Y(\mathbf{y}|\mathbf{x})$ , while the signal observed by an unaware receiver is an instance of  $q_Z(\mathbf{z}|\mathbf{x})$ . Aware receivers use the deterministic decoder  $\varphi : \mathbb{R}^T \rightarrow \mathcal{S} \times \mathcal{T}$  while unaware receivers use  $\psi : \mathbb{R}^P \rightarrow \mathcal{S}$ . See Figure 2.5.

Csiszar [24] showed that when the coding/decoding can be described by a Markov chain  $U \rightarrow V \rightarrow X \rightarrow YZ$ , the tag can be transmitted in perfect secrecy when the rates satisfy

$$0 \leq R_t \leq I(V; Y|U) - I(V; Z|U) \tag{2.46}$$

$$0 \leq R_s \leq \min[I(U; Y), I(U; Z)] \tag{2.47}$$

where  $R_s$  is the rate of the message transmission and  $R_t$  is the rate of the tag. Note that the rate of the message and tag transmissions are restricted by the disparities of Bob and Eve’s channels (Note the different probability distributions for  $Y$  and  $Z$ ).

When the tag is passed in perfect secrecy, Eve cannot gain any knowledge of the key and hence no attack can succeed above the false alarm probability.

Csiszar’s result relies on the assumption that Eve has a worse channel than Bob.

## 2.6.2 Unconditionally Secure Authentication

Wegman and Carter showed how universal hash functions may be used to achieve unconditionally secure authentication. However, the authentication is unconditionally secure for only a few messages.

*Definition [25]:* A class of hash functions  $\mathcal{H} : \mathcal{A} \rightarrow \mathcal{B}$  is strongly universal<sub>2</sub> if for all distinct  $a_1, a_2 \in \mathcal{A}$  and for all  $b_1, b_2 \in \mathcal{B}$ , the number of functions  $h \in \mathcal{H}$  for which both  $h(a_1) = b_1$  and  $h(a_2) = b_2$  hold is  $|\mathcal{H}|/|\mathcal{B}|^2$ .

It follows that a strongly universal<sub>2</sub> class of functions is also universal<sub>2</sub>.

*Definition [25]:* A class  $\mathcal{G}$  of functions  $g : \mathcal{A} \rightarrow \mathcal{B}$  is universal<sub>2</sub> if for any distinct  $a_1, a_2 \in \mathcal{A}$ , the probability that  $g(a_1) = g(a_2)$  holds is at most  $1/|\mathcal{B}|$  when  $g$  is chosen at random from  $\mathcal{G}$  according to the uniform distribution.

Now if we use the secret key to select a hash function from a strongly universal<sub>2</sub> set of hash functions, we can use it to have unconditionally secure authentication if we use the hash function to generate the tags. Now an impersonation attack has success probability  $p_{imp} = 1/|\mathcal{B}|$  because  $\mathcal{H}$  is universal<sub>2</sub>, and hence the probability that a randomly chosen message will have the correct tag is  $1/|\mathcal{B}|$  by definition.

Similarly, a substitution attack has success probability  $p_{sub} = 1/|\mathcal{B}|$  because  $\mathcal{H}$  is strongly universal<sub>2</sub>, and hence it follows from the definition that the probability that a previously seen message will map to the correct tag is  $1/|\mathcal{B}|$ .

We see that with a secret key, unconditionally secure authentication is possible. However, the above scheme with strongly universal<sub>2</sub> classes of hash function has a major weakness: it may only be used to authenticate 2 messages (the subscript <sub>2</sub> is indicative of the 2-message collision properties) with perfect security.

Allowing the adversary 2 or more (message,tag) pairs will reveal information about the hash function used, and hence make the attacks more probable.

Now we consider the possibility of authentication when Eve has non-zero knowledge of the secret key. First we introduce the notion of Renyi entropy, defined as

$$H_2(S) = -\log_2 \sum_{s \in \mathcal{S}} p_S(s)^2 \quad (2.48)$$

It is helpful to note that  $\sum_{s \in \mathcal{S}} p_S(s)^2$  is the collision probability of the random variable  $S$  - the probability that two independent realizations will be the same. This concept will be particularly useful in the context of attacks on authentication: attacks are successful when valid and invalid tags coincide. We can express Renyi entropy and use Jensen's inequality to note that  $H_2(S) = -\log_2 E[p_S(S)] \leq -E[\log_2 p_S(S)] = H(S)$

*Theorem [25]:* Let  $S$  be a binary string of (even) length  $n$ . Assume that  $S$  is used by two parties as the key in the authentication scheme based on strongly universal hashing with respect to the strongly universal<sub>2</sub> class

$$\mathcal{H} = \{h_{ab} : (a, b) \in (GF(2^N))^2\} \quad (2.49)$$

$$h_{ab}(x) := ax + b \quad (2.50)$$

and that an adversary knows a random variable  $Z$  jointly distributed with  $S$  according to some probability distribution. The adversary has no further information about  $S$ . Let

$$H_2(S|Z = z) \geq (1/2 + R) * N \quad (2.51)$$

for a particular  $z$  in the range  $\mathcal{Z}$  of  $Z$ . Then the probabilities of successful impersonation given  $Z = z$  are upper-bounded by

$$p_{imp} \leq 2^{-RN/2} \quad (2.52)$$

and

$$p_{sub} \leq 3 * 2^{-RN/4} \tag{2.53}$$

respectively. This gives upper bounds on the attack success when the Renyi entropy of the key are known.

## 2.7 Conclusion

We have introduced the scenarios for which we will design our authentication method. Because of the non-zero capacity headroom, there is leeway to insert authentication tags at the physical layer that is currently unexploited. We have also introduced and discussed the three key metrics that will be used to measure the performance of the new authentication method: stealth, robustness, and security.

## Chapter 3

### Physical Layer Authentication Framework

#### 3.1 Overview of Contributions

In this chapter we introduce the framework to describe physical layer authentication systems [26]. We begin with single-carrier systems and analyze the stealth, robustness, and security of such systems. We identify the authentication power as a major parameter and we introduce methods of improving the metrics.

We give a preview of the results as follows.

- We introduce a novel authentication system at the physical layer that has low complexity (Section 3.2).
- The authentication can be spread over multiple messages to improve robustness (Section 3.2.4).
- The transmission power dedicated to the authentication is a major determining factor in the performance of the system. Lower-powered authentication trades robustness for improved stealth and security (Section 3.3).
- Error correcting codes are useful not only for the message reliability but are also essential for authentication since only error-free messages should be authenticated (Section 3.3.2).

- The framework is extended to fast-fading AR-1 channels and demonstrated to be effective in high SNR regimes (Section 3.4).

## 3.2 Framework

### 3.2.1 Signal Model

The sender wants to transmit a message to the receiver so that it can be recovered and understood. The sender codes and modulates the message to protect against errors when it is passed through a random channel.

The signal is transmitted in frames of length  $N^f$  composed of message and authentication symbols. Assume the signals are i.i.d. and thus we drop any time indices. Denote a frame by the column vector  $\mathbf{x}$  with complex entries  $\{x_1, \dots, x_{N^f}\}$ . Let each entry have mean 0 and variance  $\sigma_x^2$

$$E[x_m] = 0 \tag{3.1}$$

$$E|x_m|^2 = \sigma_x^2 \tag{3.2}$$

$$E|\mathbf{x}|^2 = E\left[\sum_m |x_m|^2\right] = N^f \sigma_x^2 \tag{3.3}$$

Each frame is formed by superimposing the coded message  $\mathbf{s}$  and authentication tag  $\mathbf{t}$ :

$$\mathbf{x} = \rho^s \mathbf{s} + \rho^t \mathbf{t} \tag{3.4}$$

where  $\rho^s, \rho^t$  are scaling terms in  $[0, 1]$  used to enforce the constraints (3.1) and (3.2). Assuming that the message and tag are uncorrelated:

$$E[\mathbf{s}^H \mathbf{t}] = 0 \tag{3.5}$$

it follows that

$$E[s_m] = 0, E|s_m|^2 = \sigma_x^2 \quad (3.6)$$

$$E[t_m] = 0, E|t_m|^2 = \sigma_x^2 \quad (3.7)$$

$$(\rho^s)^2 + (\rho^t)^2 = 1 \quad (3.8)$$

The message  $\mathbf{s}$  is generated from the uncoded message block  $\mathbf{b}$ , and the authentication is generated from  $\mathbf{b}$  and a shared secret key  $k \in \mathcal{K}$  where  $|\mathcal{K}| = K$ :

$$\mathbf{s} = f_e(\mathbf{b}) \quad (3.9)$$

$$\mathbf{t} = g(\mathbf{b}, k) \quad (3.10)$$

The encoding function  $f_e(\cdot)$  encapsulates any coding, modulation, or pulse shaping that may be used. The corresponding decoding function  $f_d(\cdot)$  is used at the receiver and satisfies

$$\mathbf{b} = f_d(f_e(\mathbf{b})) \quad (3.11)$$

for all possible inputs  $\mathbf{b}$  of  $f_e(\cdot)$ . The tag generating function  $g(\cdot)$  is assumed to be one-way, i.e., it is easy to calculate  $\mathbf{t}$  given  $\mathbf{b}$  and  $k$ , but hard to find  $k$  given  $\mathbf{t}$  and  $\mathbf{b}$ . The usefulness of this property will be discussed in Section 3.3.3. Further, it is also collision resistant so that it is hard to find  $x \neq y$  such that  $g(x, k) = g(y, k)$ .

### 3.2.2 Channel Model and Estimation

We assume a Rayleigh block fading channel so that different message blocks experience independent fades that are constant for the duration of each block. While this may not be strictly true, is a reasonable assumption for slow fading channels.

Consider a channel realization  $h$  which is a complex Gaussian variable with mean 0 and variance  $\sigma_h^2$ . The receiver observes the frame

$$\mathbf{y} = h\mathbf{x} + \mathbf{w} \quad (3.12)$$

where  $\mathbf{w} = \{w_1, \dots, w_{N_f}\}$  and  $w_m \sim N(0, \sigma_w^2)$  is complex white Gaussian noise. The average SNR is  $\bar{\gamma} = \sigma_h^2 \sigma_x^2 / \sigma_w^2$ . The SNR experienced by each block  $\gamma$  is Rayleigh distributed with density

$$p(\gamma) = \frac{1}{\bar{\gamma}} e^{-\gamma/\bar{\gamma}} \quad (3.13)$$

When the SNR  $\gamma$  falls below a certain threshold, say  $\gamma^0$ , the observed frame becomes unacceptably corrupted. The outage probability is the fraction of time that this occurs. The outage probability  $P_{out}$  can be fixed by setting the transmission power  $\sigma_x^2$  to change the average SNR  $\bar{\gamma}$

$$P_{out} = \int_0^{\gamma^0} p(\gamma) d\gamma = 1 - e^{-\gamma^0/\bar{\gamma}} \quad (3.14)$$

$$\bar{\gamma} = \frac{-\gamma^0}{\ln(1 - P_{out})} \quad (3.15)$$

Pilot symbols are typically used to aid in channel estimation, and we insert them in the middle of the block. (We use this as a representative pilot scheme, however, we emphasize our framework easily generalizes to other cases). For the pilot symbols  $\mathbf{p}$  and their observations  $\mathbf{y}_p$ , the MMSE channel estimate is simply

$$\hat{h} = \frac{1}{|\mathbf{p}|^2} \mathbf{p}^H \mathbf{y}_p \quad (3.16)$$

$$= \frac{1}{|\mathbf{p}|^2} \mathbf{p}^H (h\mathbf{p} + \mathbf{w}_p) \quad (3.17)$$

$$= h + \eta \quad (3.18)$$

where  $(\cdot)^H$  is the Hermitian transpose and  $\eta$  is a complex Gaussian variable with zero mean and variance  $\sigma_\eta^2 = \sigma_w^2 / |\mathbf{p}|^2$ . We consider the pilot symbols as separate from frame  $\mathbf{x}$ , i.e., the authentication framework does not modify the pilots used for channel estimation.

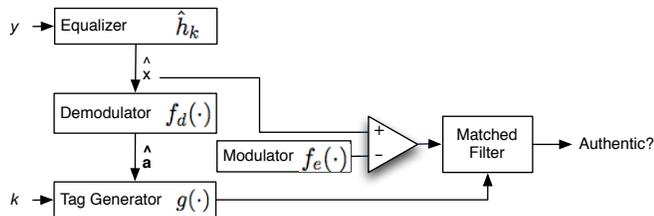


Figure 3.1: Block diagram of aware receiver.

### 3.2.3 Message Recovery

A block diagram of the aware receiver is found in Figure 3.1.

The receiver uses its channel estimate to estimate the transmitted message signal

$$\hat{\mathbf{s}} = \frac{\hat{h}^*}{|\hat{h}|^2} \mathbf{y} \quad (3.19)$$

It then uses  $f_d(\cdot)$  to recover the message symbols

$$\hat{\mathbf{b}} = f_d(\hat{\mathbf{s}}) \quad (3.20)$$

For analysis, we assume that the message and tag signals  $\mathbf{s}, \mathbf{t}$  are both modulated using 4-QAM. The tag constellation is superimposed on each message symbol to form the constellation shown in Figure 3.2. In the literature this is called the 4/16 hierarchical QAM constellation. Note that the constellation has 16 symbols; each of these symbols signal the message symbol (which quadrant) and the tag symbol (which point within the quadrant). Of course, many different superposition schemes may be considered, and are discussed in Section 3.2.5.

#### BER Calculation

To calculate the uncoded BER for a hierarchical QAM constellation, we must know the distance between the symbols as well as the noise power. Let  $2d_1$  be the minimum distance between any two (fictitious) message points,  $2d_2$  the

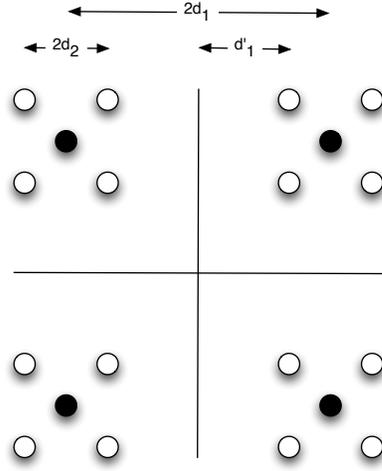


Figure 3.2: 4/16 hierarchical QAM constellation. The 16 white circles indicate the constellation points; the black circles indicate the original 4-QAM constellation used for the message symbols.

minimum distance between any two tag points within the same quadrant, and  $2d'_1$  be the minimum distance between points in adjacent quadrants (see Figure 3.2). We first give the exact BER expressions for unit-energy 4/16 constellations and then apply them to the authentication signals. In general, any number of QAM constellations may be superimposed on each other. The general expressions for the BER of general hierarchical QAM constellations are given in [27].

The BER of the message bit is [27]

$$p^s = \frac{1}{2} \left( \frac{1}{2} \operatorname{erfc} \frac{d'_1}{\sqrt{N_0}} + \frac{1}{2} \operatorname{erfc} \frac{d'_1 + 2d_2}{\sqrt{N_0}} \right) \quad (3.21)$$

$$= \frac{1}{4} (\Psi(1, 0) + \Psi(1, 2)) \quad (3.22)$$

where  $N_0$  is the noise power, while the BER of the tag bit is

$$p^t = \frac{1}{2} \left( \operatorname{erfc} \frac{d_2}{\sqrt{N_0}} + \frac{1}{2} \operatorname{erfc} \frac{2d'_1 + d_2}{\sqrt{N_0}} - \frac{1}{2} \operatorname{erfc} \frac{2d'_1 + 3d_2}{\sqrt{N_0}} \right) \quad (3.23)$$

$$= \frac{1}{4} (2\Psi(0, 1) + \Psi(2, 1) + \Psi(2, 3)) \quad (3.24)$$

The function  $\Psi(\cdot)$  depends on the channel distribution, and for the Rayleigh channel with unit energy constellations it is [27]

$$\Psi(a, b) = 1 - \sqrt{\frac{(ad'_1 + bd_2)^2 \gamma}{1 + (ad'_1 + bd_2)^2 \gamma}} \quad (3.25)$$

The noise power  $N_0$  is the average noise power for a unit variance channel. For perfect channel information, it is  $1/\gamma$ , while for MMSE estimation it is  $1/\gamma^{lmmse} > 1/\gamma$ .

The effective signal power is  $\sigma_h^2 \sigma_x^2$ . In order to use Equations (3.21) and (3.23), the constellation needs to be unit energy. Thus we scale

$$\tilde{\rho}^s = \rho^s / \sigma_h \sigma_x \quad (3.26)$$

$$\tilde{\rho}^t = \rho^t / \sigma_h \sigma_x \quad (3.27)$$

$$N_0 = \sigma_w^2 / \sigma_h^2 \sigma_x^2 = 1/\gamma \quad (3.28)$$

and calculate the following parameters:

$$d_1 = \tilde{\rho}^s \sqrt{2} \quad (3.29)$$

$$d_2 = \tilde{\rho}^t \sqrt{2} \quad (3.30)$$

$$d'_1 = d_1 - d_2 \geq 0 \quad (3.31)$$

We calculate the BER of the message and tag bits by using these values in equations (3.21) and (3.23) respectively. Note that equations (3.29)-(3.31) require that  $\rho^s > \rho^t$ . Otherwise, the decision boundaries will overlap.

## Outage Probability

As mentioned previously, the message  $\mathbf{s}$  is a codeword corresponding to the uncoded message  $\mathbf{b}$  according to the encoding function  $f_e(\cdot)$ . We are interested in the ability of Alice to recover  $\mathbf{b}$  without error.

In fading channels, the importance of minimum Euclidean distance between codewords is supplanted by the minimum Hamming distance [28]. Suppose that Alice and Bob use BCH codes to protect the messages. The minimum distance is lower-bounded by the BCH bound.

Consider the  $(n, k, t)$  BCH code. It encodes  $k$  message bits into  $n$  code bits and is able to recover from up to  $t$  errors. With bit interleaving, the symbol errors can be assumed independent, and thus the probability that there are at most  $t$  errors in  $n$  bits is given by the binomial cdf

$$P = \sum_{i=0}^t \binom{n}{i} (p^s)^i (1 - p^s)^{n-i} \quad (3.32)$$

Thus given the message bit error  $p^s$  from equation (3.21), the probability that the BCH code  $(n, k, t)$  can recover the message without error is  $P$ . The probability of message error  $p^s$  is determined by the channel realization  $h$ . That is, the BER requirement may be viewed as a minimum SNR requirement since the BER is monotonic versus SNR.

Suppose that the system uses the code  $BCH(n, k, t)$ . Given the SNR  $\gamma^0$ , the message is recovered correctly with probability  $P$ . When the SNR falls below  $\gamma^0$  the recovery probability also falls below  $P$  and we say that it is an outage. For the Rayleigh channel, the message outage for a particular SNR requirement  $\gamma^0$  is given by equation (3.14).

### 3.2.4 Tag Detection

In addition to recovering the message, the aware receiver also decides on the authenticity of the signal. If the receiver decides that the observation demonstrates sufficient knowledge of the key, then it authenticates the sender. Otherwise, the signal is not authenticated. In this section we introduce the method of authentication and show that it is low-complexity.

After estimating the channel, the receiver proceeds to perform message estimation and obtains  $\hat{\mathbf{s}}$ . With the secret key, it can generate the estimated tag  $\hat{\mathbf{t}}$  using equation (3.10) and look for it in the residual  $\mathbf{r}$ .

$$\hat{\mathbf{t}} = g(\hat{\mathbf{b}}, k) \quad (3.33)$$

$$\mathbf{r} = \frac{1}{\rho^t} \left( \frac{\hat{h}^*}{|\hat{h}|^2} \mathbf{y} - \rho^s f_e(\hat{\mathbf{b}}) \right) \quad (3.34)$$

The receiver performs a threshold test with hypotheses

$$H_0 : \quad \hat{\mathbf{t}} \text{ is not present in } \mathbf{y} \quad (3.35)$$

$$H_1 : \quad \hat{\mathbf{t}} \text{ is present in } \mathbf{y} \quad (3.36)$$

We obtain our test statistic  $\tau$  by match filtering the residual with the estimated tag:

$$\tau = \Re(\hat{\mathbf{t}}^H \mathbf{r}) \quad (3.37)$$

where  $\Re(\cdot)$  takes the real component of its argument.

The decision of authenticity  $\delta$  is made according to

$$\delta = \begin{cases} 0 & \tau < \tau^0 \\ 1 & \tau \geq \tau^0 \end{cases} \quad (3.38)$$

The threshold  $\tau^0$  of this test is determined for a false alarm probability  $\alpha$  according to the distribution of  $(\tau|H_0)$ .

The decision of authenticity can be made after only a few operations: re-generation of the tag, calculation of the residual, and correlation of the tag and residual. These are all simple operations (including the tag generation since it is a one-way function as assumed in Section 3.2.1), and therefore the authentication is low-complexity.

### Determination of $\tau^0$

In order to limit the false alarm probability  $\alpha$ , we calculate the threshold  $\tau^0$  such that  $P(\tau > \tau^0 | H_0) \leq \alpha$ . There are two main cases where a false alarm can occur: when the observation contains no tag at all or when the observation contains an incorrect tag.

First, we consider the structure of the residual  $\mathbf{r}$ . Using equation (3.12), we have

$$\frac{\hat{h}^*}{|\hat{h}|^2} \mathbf{y} = \frac{\hat{h}^*}{|\hat{h}|^2} (h\mathbf{x} + \mathbf{w}) \quad (3.39)$$

$$= \frac{\hat{h}^*}{|\hat{h}|^2} (\hat{h}\mathbf{x} - \eta\mathbf{x} + \mathbf{w}) \quad (3.40)$$

$$= \mathbf{x} + \frac{\hat{h}^*}{|\hat{h}|^2} (-\eta\mathbf{x} + \mathbf{w}) \quad (3.41)$$

$$= \mathbf{x} + \hat{\mathbf{w}} \quad (3.42)$$

where  $\hat{\mathbf{w}}$  is a vector of complex AWGN with zero mean and variance  $\sigma_{\hat{\mathbf{w}}}^2 = (\sigma_{\eta}^2 \sigma_x^2 + \sigma_w^2) / |\hat{h}|^2$ . Thus the residual is Gaussian distributed.

In the following we assume that the message is received without error ( $\hat{\mathbf{b}} = \mathbf{b}$ ) since only correct messages should be authenticated. When a message is received with error, it will generate an incorrect tag (addressed in case 2 below) with high probability due to the collision resistance of  $g(\cdot)$  (Section 3.2.1). This should be interpreted as follows: when a message is received with error, it will be

authenticated with the false alarm probability  $\alpha$ , and when it is received correctly, it will be authenticated with the detection probability  $P^a$ .

Case 1: the transmitted signal does not contain any tag (i.e.,  $\mathbf{x} = \mathbf{s}$ ). Then

$$\tau|H_0 = \Re(\hat{\mathbf{t}}^H \mathbf{r}) \quad (3.43)$$

$$= \Re\left(\frac{1}{\rho^t} \hat{\mathbf{t}}^H ((1 - \rho^s) \mathbf{s} + \hat{\mathbf{w}})\right) \quad (3.44)$$

$$= \frac{1 - \rho^s}{\rho^t} \Re(\hat{\mathbf{t}}^H \mathbf{s}) + v \quad (3.45)$$

where  $v$  is real Gaussian variable with zero mean and variance  $\sigma_v^2 = |\mathbf{t}|^2 \sigma_{\hat{\mathbf{w}}}^2 / 2(\rho^t)^2$ .

The term  $\Re(\hat{\mathbf{t}}^H \mathbf{s})$  is a sum of  $N^f$  i.i.d. variables and is well approximated by a Gaussian distribution when  $N^f$  is large (central limit theorem). From equation (3.5) the mean is zero and the variance  $\sigma_{ts}^2$  depends on the symbol constellations. For example, if the message and tag are QPSK symbols, the variance of  $\Re(\hat{\mathbf{t}}^H \mathbf{s})$  is  $\sigma_{ts}^2 = N^f * \frac{1}{4}(1 + 1 + 0 + 0) = \frac{N^f}{2}$ . Thus the test statistic  $\tau$  is Gaussian with zero mean and variance

$$\sigma_\tau^2 = \left(\frac{1 - \rho^s}{\rho^t}\right)^2 \sigma_{ts}^2 + \sigma_v^2 \quad (3.46)$$

Case 2: the transmitted signal contains a tag different from  $\hat{\mathbf{t}}$ . Then

$$\tau|H_0 = \Re(\hat{\mathbf{t}}^H \mathbf{r}) \quad (3.47)$$

$$= \Re\left(\frac{1}{\rho^t} \hat{\mathbf{t}}^H (\rho^t \mathbf{t} + \hat{\mathbf{w}})\right) \quad (3.48)$$

$$= \Re(\hat{\mathbf{t}}^H \mathbf{t}) + v \quad (3.49)$$

where  $v$  is real Gaussian variable with zero mean and variance  $\sigma_v^2 = |\mathbf{t}|^2 \sigma_{\hat{\mathbf{w}}}^2 / 2(\rho^t)^2$ .

The term  $\Re(\hat{\mathbf{t}}^H \mathbf{t})$  is a sum of  $N^f$  i.i.d. variables and is well approximated by a Gaussian distribution when  $N^f$  is large (central limit theorem). From equation (3.5) the mean is zero and the variance  $\sigma_{tt}^2$  depends on the tag symbol constellation. Thus the test statistic  $\tau$  is Gaussian with zero mean and variance

$$\sigma_\tau^2 = \sigma_{tt}^2 + \sigma_v^2 \quad (3.50)$$

Without priors, the threshold is calculated based on the worst case distribution from either case 1 or 2. Since both are zero mean Gaussian distributions, the worst case has the largest variance  $\tilde{\sigma}_\tau^2$  from equations (3.46) and (3.50).

$$\tau^0 = \arg \min_{\tau} \Phi(\tau/\tilde{\sigma}_\tau) \geq 1 - \alpha \quad (3.51)$$

where  $\Phi(\cdot)$  is the standard Gaussian cumulative distribution function.

### Probability of Authentication

The performance of the authentication is directly tied to the performance of the message; authentication will occur only when the message is correctly received. This is logical because distorted messages should not be authenticated.

When Bob generates the correct tag ( $\hat{\mathbf{t}} = \mathbf{t}$ ), the test statistic is

$$\tau|H_1 = |\mathbf{t}|^2 + v. \quad (3.52)$$

The probability of detection is

$$P^a(\gamma) = 1 - \Phi((\tau^0 - |\mathbf{t}|^2)/\sigma_v). \quad (3.53)$$

The overall probability of detection is

$$P^a = E[P^a] = \int P^a(\gamma)p(\gamma)d\gamma \quad (3.54)$$

where  $p(\gamma)$  is the probability density of  $\gamma$  given in equation (3.13).

When the noise is i.i.d., there is no benefit in restricting the number of tagged symbols since the authentication depends on  $|\mathbf{t}|^2$  alone. That is, concentrating the tag power in a few symbols does not increase the detection probability. However, that strategy may make the tag more noticeable and therefore less stealthy. The stealth of the tag depends on the characteristics of the noise. For example, if the noise is impulsive (distribution is heavy-tailed), it is more stealthy to have the tag also be impulsive (and spread over few symbols). On the other hand, if the noise is Gaussian, it is more stealthy to have the tag spread over more symbols.

## Outage Probability

The authentication decision is useful when the false alarm probability is very low. Of nearly equal importance is the probability that valid messages are authenticated. Again, we note that the detection probability is a monotonic function of SNR, so we map the detection requirement to an SNR requirement. Thus when we fix the false alarm probability  $\alpha$  and require the detection probability to exceed a given  $P^a$ , this is equivalent to requiring a minimum SNR  $\gamma^0$ . When the SNR falls below  $\gamma^0$  we say that the authentication is in an outage. For the Rayleigh channel, the authentication outage for a particular SNR requirement  $\gamma^0$  is given by equation (3.14).

## Authentication over Multiple Frames

In the following we consider how the robustness of the authentication may be improved.

One possible method of improving the detection probability is to raise the average SNR  $\bar{\gamma}$  by increasing the transmission power  $\sigma_x^2$ . This lowers the probability of unsuitably low SNRs, but is not always feasible due to physical constraints. Alternatively, we may extend the authentication process to consider many frames together instead of each frame separately. Because we are assuming a Rayleigh block fading channel model, each frame experiences independent fades, and conditioned on the authenticity of the signal the authentication decisions are independent events as well.

Suppose that authentication is transmitted over each frame. Let  $x = \sum_{i=1}^K \delta_i$  to tally the number of detected tags in  $K$  blocks. When no tag is sent, the

probability of detecting more than  $k_0$  tags is

$$p(x > k_0 | H_0) = \sum_{i=k_0+1}^K B(i; K, \alpha) \quad (3.55)$$

where  $B(x; n, p)$  is the binomial probability mass function of getting exactly  $x$  successes in  $n$  identical and independent trials with probability of success  $p$ . For the extended test, we compare  $x$  with a threshold  $k_0$  that is set so that the false alarm probability does not exceed the new false alarm probability  $\alpha_K$

$$k_0 = \arg \min_j \left[ \sum_{i=j+1}^K B(i; K, \alpha) < \alpha_K \right] \quad (3.56)$$

The Neyman-Pearson test gives the probability of deciding  $H_1$  as:

$$\delta_K = \begin{cases} 1 & x < k_0 \\ \pi & x = k_0 \\ 0 & x > k_0 \end{cases} \quad (3.57)$$

where  $\pi$  is the randomization of the detection rule and is given by

$$\pi = \frac{\alpha_K - p(x > k_0 | H_0)}{p(x = k_0 | H_0)} \quad (3.58)$$

For a randomly selected group of  $K$  tagged signal blocks, the probability of correctly deciding  $H_1$  is simply

$$p(x > k_0 | H_1) = (1 - \pi)B(k_0; K, P^a) + \sum_{i=k_0+1}^K B(i; K, P^a) \quad (3.59)$$

where  $P^a$  is the probability of detection for a randomly observed block (see equation (3.54)).

There is a fundamental tradeoff between robustness and security. When a scheme is made more robust in this manner, we are allowing more errors to be made in the tag detection before rejecting an authentic signal. However, this gives the adversary more opportunity to inject malicious blocks that may be accepted as authentic. We will discuss the security issues in Section 3.3.3.

### 3.2.5 Superposition Methods

There are many ways to structure the tag for superpositioning - we are not restricted to the use of hierarchical constellations as considered in the previous sections. The content of the signal dictates the effect that the tag has on message recovery.

In the previous section we considered hierarchical QAM constellations for ease of implementation as well as clarity of discussion. At high SNR, it becomes easy for the receiver to distinguish the constellation points. Therefore, for stealth purposes we can also consider the use of arbitrarily distributed tags rather than using fixed modulation points. Suppose that the additive noise is Gaussian so that we add Gaussian tags. While the tags become more stealthy since they appear more noise-like, they also add more interference to the message symbols because the magnitude of the tag is no longer strictly limited. Figure 3.3 shows the message BER when the tags are either 4QAM or Gaussian. Clearly, the hierarchical constellation is a much better choice in terms of message BER; the Gaussian tags cause the message BER to hit an error floor. This can be explained because the effective SNR can be written

$$\tilde{\gamma} = \frac{\rho^s \sigma_x^2}{\sigma_w^2 + (1 - \rho^s) \sigma_x^2} \quad (3.60)$$

$$\lim_{\sigma_w^2 \downarrow 0} \tilde{\gamma} = \frac{\rho^s}{1 - \rho^s} \quad (3.61)$$

where again  $\sigma_x^2, \sigma_w^2$  are the signal and noise powers, respectively. Note that even as the noise power drops to zero, the effective SNR  $\tilde{\gamma}$  does not go to infinity because  $(1 - \rho^s)$  of the signal power is dedicated to the tag. This illustrates that there is an inherent tradeoff between stealth and security.

In general, the signal may be viewed in terms of a basis decomposition. There are a large variety of bases to choose from, but some are more appropriate than

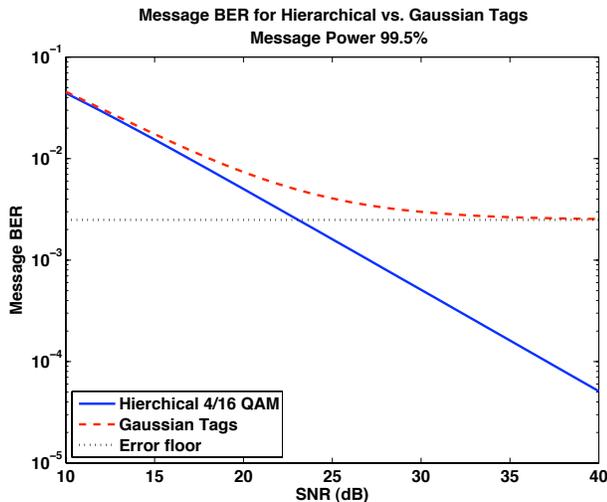


Figure 3.3: Message BER are compared for hierarchical 4/16 QAM constellation versus normally distributed tags. The error floor for the Gaussian tags is determined by the message/tag power allocation  $\rho^s$ .

others to describe the structure and impact of the authentication. Fundamentally, signals cannot be limited in both time and frequency: time-limited signals have power over infinite frequencies while band-limited signals extend for all time. However, real signals have energy concentrated in finite time intervals and frequency bands<sup>1</sup>. Bases that do not consider such tradeoffs (e.g., Fourier) offer either very poor time or frequency resolution. Authentication tags that are composed using such alphabets are not appropriate because they are neither stealthy nor secure (Section 3.3). Furthermore, they offer limited degrees of freedom in which to signal the authentication, which ideally is drawn from very large spaces.

Multiresolution techniques such as wavelet transforms are a more appropriate approach since they give basis coefficients which are localized in both time and frequency (Figure 3.4). This allows the system designer to selectively place the

<sup>1</sup>If a signal concentrates most  $(1 - \epsilon)$  of its energy in the time interval  $(-T/2, T/2)$  and band  $(-\Omega, \Omega)$ , the approximate dimension of the process is  $2\Omega T$  as  $\Omega T$  grows large.

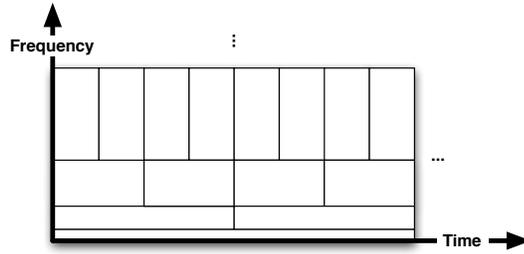


Figure 3.4: The wavelet tiling of the time-frequency plane. The wavelet basis yields a multi-resolution view of the signal that trades time resolution for frequency resolution.

authentication tag energy where it can be simultaneously hidden from adversaries and non-intrusive on the message recovery. One issue that arises is the potential bandwidth expansion introduced by the authentication tag. With the appropriate choice of basis, it is easy to control the bandwidth of the resulting signal. For example, with the wavelet basis it is simple to place energy within the bandwidth constraints of the message signal by restricting the use of high-frequency basis functions.

The view of the signal via a basis decomposition allows the framework presented above to describe a large class of superimposed authentication systems. No longer restricted to the view of the signal on a symbol-by-symbol basis, the basis view expands the richness of the space to describe bandwidth shaping and other realities in which the authentication can be hidden.

### 3.3 Metric Evaluation

We illustrate the tradeoffs of the scheme by studying an example system where the message and tag symbols are simultaneously modulated using a hierarchical 4/16-QAM constellation (Figure 3.2). The Rayleigh block fading channel is esti-

Table 3.1: Simulation parameters for the single carrier, Rayleigh block fading case

Channel Model	Rayleigh block fading
Noise Model	AWGN
Channel Estimate Method	ML
# Pilot Symbols	16 per frame
Frame Length	256, 512, or 1024 symbols
False Alarm Probability	$10^{-7}$
# Monte Carlo Samples	$2^{15}$

mated using a 16-symbol pilot sequence within 256-symbol frames. We assume that the receiver experiences white Gaussian noise. The Monte Carlo simulation parameters are summarized in Table 3.1.

A range of false alarm probabilities was simulated, and the pictured false alarm probability was chosen to give an example of a reasonable operating point. We note that as the false alarm probability becomes smaller, the power of the authentication test does not change much because the tails of the distribution under  $H_0$  (3.35) are very small. With suitably low false alarms, the authentication decisions may be trusted with high probability.

### 3.3.1 Stealth

We recall the two aspects of stealth: the presence and the impact of the authentication tag. We discuss the two aspects in turn.

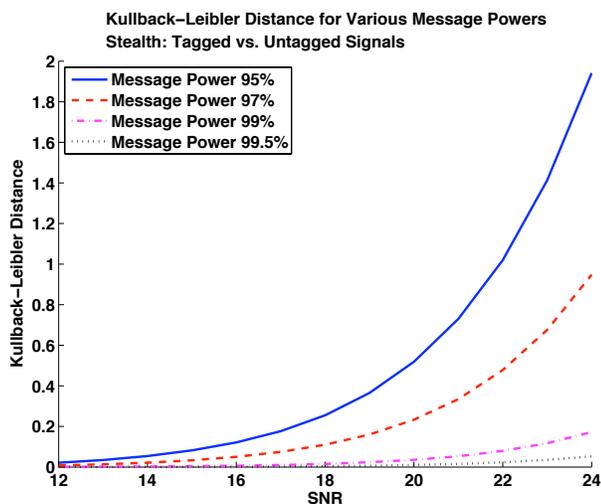


Figure 3.5: Stealth: KL Distance between tagged and untagged signals for various message powers in AWGN. Higher message power makes the tag more hidden and hence leads to better stealth.

### Presence

The presence of a signal is difficult to detect when its distribution is close to the baseline noise distribution. This fact emphasizes our view of noise as beneficial towards stealth: our scheme cannot be stealthy in a noiseless environment. The 'closeness' of two distributions is measured by the Kullback-Leibler distance and is shown in Figure 3.5 for various message powers. We note that for all  $\rho^s$  the scheme is hidden for low to medium SNRs (below 18dB) but is increasingly easy to detect for higher SNRs. The intuition is that as the noise decreases, the slight perturbations caused by the authentication become increasingly discernible. However, it is also clear that the higher the message power (and hence the lower the authentication power), the more well-hidden the authentication becomes.

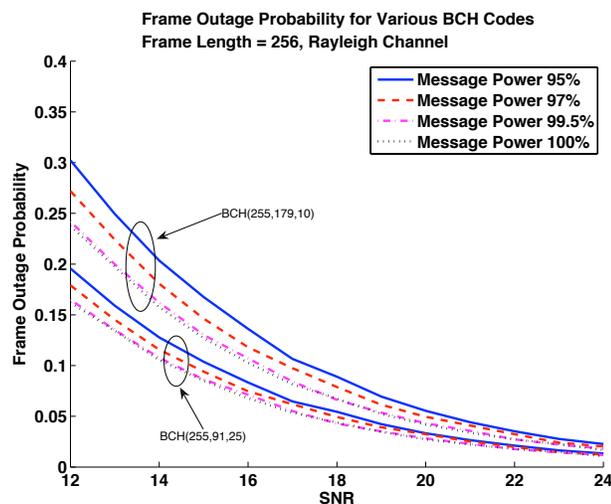


Figure 3.6: Stealth: Frame outage probabilities for various BCH codes in the Rayleigh fading channel. Higher message power admits lower frame outage probability and hence better stealth.

### Impact

The impact of the authentication on the unaware receiver can be measured easily by the receiver's message outage probability.

With a reduction in SNR, we examine the increase in outage probability for various powers of error correcting codes. Figure 3.6 shows the outage probability of the frames (an outage occurs when a message is not recovered correctly). Clearly more powerful codes decrease the outage probability. They also diminish the impact of the authentication upon the message. For the BCH(255,179,10) code, reducing the message power to 95%, 97%, and 99.5% increases the outage probability by 0.1%, 1.1%, and 2.2%, respectively, at 18 dB. However, for the stronger BCH(255,91,25) code, reducing the message power to 95%, 97%, and 99.5% increases the outage probability by 0.1%, 0.5%, and 1.0%, respectively, at 18 dB.

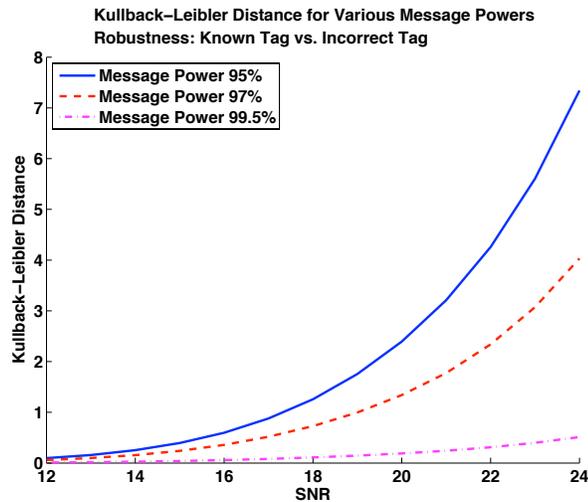


Figure 3.7: Robustness: KL Distance between correct and incorrect tags for various message powers in AWGN. Lower message power makes the tag more easily detected to Bob and hence better robustness.

### 3.3.2 Robustness

The robustness of the authentication is given by the ability of the aware receiver to accurately determine the validity of any given message. Fundamentally, the performance is limited by the Kullback-Leibler distance between the correct and incorrect tags observed in noise. Figure 3.7 shows the KL distance for various message powers. Immediately we see that lower message powers (and hence higher tag powers) lead to greater distances and hence more reliable detections.

Since we authenticate only unmodified messages, in the following we assume that the message is received correctly. For each frame, the receiver decides the authenticity of the message using a threshold that depends on the channel estimate (Section 3.2.4). When the detection probability is not high enough, we consider the performance to be insufficient. Figure 3.8 shows the authentication outage probabilities for various message powers for false alarm  $\alpha = 10^{-7}$  and

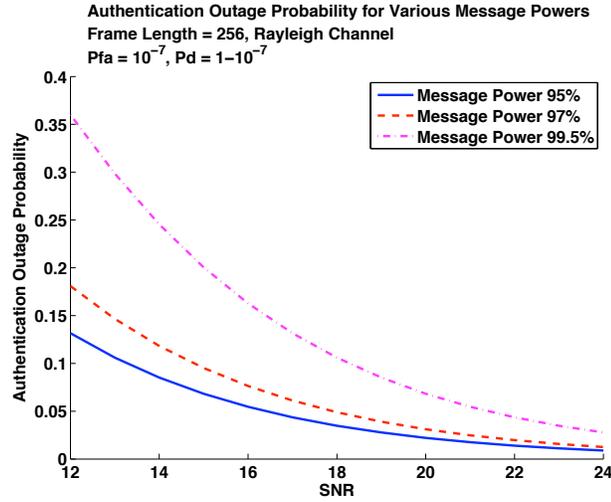


Figure 3.8: Robustness: Authentication outage probabilities for various message powers in the Rayleigh fading channel. Lower message power admits lower authentication outage and hence better robustness.

minimum detection probability  $P^a = 1 - 10^{-6}$ . Using higher tag powers improves the robustness of the authentication, though the benefit diminishes as the SNR increases.

In some situations the tag power cannot be increased further due to stealth requirements (Section 3.3.1). Another way to increase the energy of a tag is to increase its length. This also improves the robustness of the authentication, as shown in Figure 3.9.

The authentication performs well for high SNR, and the performance is improved when the tag power is increased, the tag length is increased, when the tag is coded over multiple messages, or when the message is sufficiently coded for accurate recovery. When the outage probability of the authentication is very low, it may replace conventional message authentication codes. However, in other situations it may be used to supplement the existing authentication.

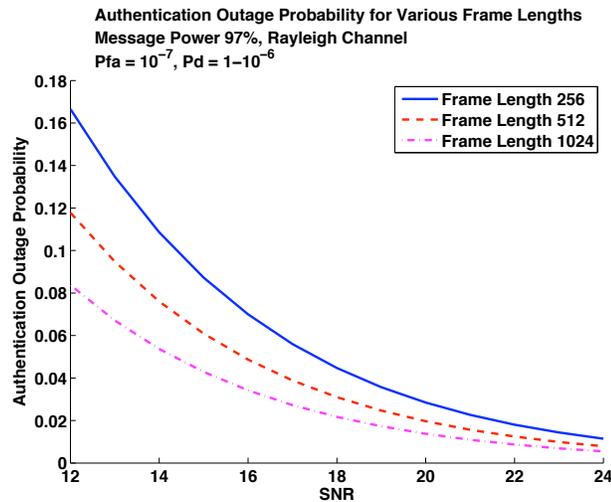


Figure 3.9: Robustness: Authentication outage probabilities for various tag lengths in the Rayleigh fading channel. Longer frames admit higher authentication probability and hence better robustness.

### 3.3.3 Security

The security of the authentication system is measured by how easily Eve can recover the secret key. Eve can try to estimate the tag and recover the key. We consider each step in order.

#### Tag Equivocation

Given a particular message  $\mathbf{s}$  and a particular tag  $\mathbf{t}$ , there is no guarantee of positive key equivocation unless the the number of keys is greater than the number of tags (by the pigeonhole principle). However, observing the tag through a noisy channel leads to positive key equivocation.

The equivocation of the authentication tag depends on the bit error rate that it is observed with. Suppose that the authentication tag  $\mathbf{t}$  is composed of  $N$  bits and is observed with i.i.d. bit errors with probability  $p^t$ . We can calculate the

tag equivocation  $H(\mathbf{t}|p^t)$  by iterating through the number of bit errors the tags can contain (between 0 and  $N$ ). The probability of observing  $n$  errors in a length  $N$  tag with bit error probability  $p^t$  is

$$Pr(p^t, n, N) = (p^t)^n(1 - p^t)^{N-n} \quad (3.62)$$

The tag bit error probability  $p^t$  depends on the system parameters as described in Section 3.2.3. For example, the system using a hierarchical 4/16-QAM constellation observed through a Rayleigh channel is parametrized with tag power  $(\rho^t)^2$  and average SNR  $\gamma$ . The resulting tag bit error probability is given in equation (3.23).

Since tags with the same number of i.i.d. bit errors have the same probability of occurring (and there are  $\binom{N}{n}$  length  $N$  tags with  $n$  errors), the tag equivocation is

$$H(\mathbf{t}|p^t) = \sum_{\mathbf{t} \in \mathcal{T}} Pr(\mathbf{t} = t|p^t) \log_2 \frac{1}{Pr(\mathbf{t} = t|p^t)} \quad (3.63)$$

$$= \sum_{n=0}^N \binom{N}{n} Pr(p^t, n, N) \log_2 \frac{1}{Pr(p^t, n, N)} \quad (3.64)$$

where  $Pr(\cdot, \cdot, \cdot)$  is defined above in equation (3.62). We note that there is an abuse of notation in this equation in that  $p^t$  is really a constant; it does not depend on the particular tag.

Figure 3.10 shows the tag equivocation for a 512-bit tag over a Rayleigh block fading channel for various message powers. Higher message powers yield higher equivocations. Note that the equivocation decreases as the SNR increases because with less noise, the tag is less stealthy and may be more easily distinguished and decoded. Thus with higher tag equivocation the key equivocation is similarly increased (Chapter 2).

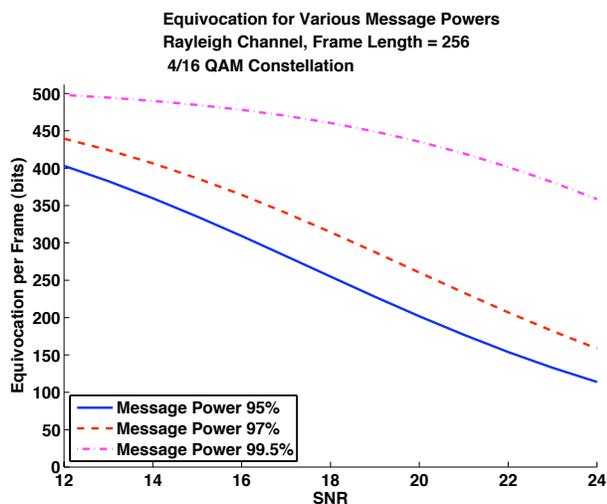


Figure 3.10: Security: Equivocation of tags for various message powers in the Rayleigh fading channel. Higher message power leads to higher equivocation and hence better security.

### Vulnerabilities

When multiple blocks are used for the authentication, the added robustness gives the adversary more opportunities to pass inauthentic blocks to Bob. The tradeoff between robustness and security is fundamental - by allowing more errors in the authentication process, Eve has a better opportunity to sneak in her own messages. However, since messages may be coded across blocks to mitigate outage effects, Eve's malicious blocks will be corrected, or more likely discarded, by the decoder. Eve must therefore be able to convince Bob to accept a stream of tagged messages, something that is very difficult when she does not know the secret key.

## 3.4 Extension to Fast-Fading Channels

Thus far we have considered the slow fading channel. A natural question that arises is how well the scheme works in fast fading channels. To tackle this ques-

tion, we introduce another channel model and the associated channel estimation algorithm. We find that the aware receiver can even improve his message recovery (by trading off delay) by treating the authentication tag as pilot symbols, and we detail the necessary changes.

### 3.4.1 Channel Model

Instead of the Rayleigh block fading channel, we use a Gauss-Markov channel model to describe fast flat fading [29]. Rather than assuming a constant fade for each block of symbols, each symbol suffers a different but correlated fade. The channel for the  $k^{\text{th}}$  symbol is

$$h_k = ah_{k-1} + u_k \tag{3.65}$$

where  $a$  is the fading correlation coefficient, and  $u_k \sim N(0, \sigma_u^2)$  where  $\sigma_u^2 = (1-a^2)\sigma_h^2$ . The fading correlation coefficient characterizes how quickly the channel fades: large values (close to unity) model slow fading channels while smaller values model fast fading channels. After passing through the channel, the receiver observes the signal  $\mathbf{y}$ :

$$y_k = h_k x_k + w_k \tag{3.66}$$

$$\mathbf{y}_i = \mathbf{h}_i \cdot \mathbf{x}_i + \mathbf{w}_i \tag{3.67}$$

where as before  $w_k \sim N(0, \sigma_w^2)$  is white Gaussian noise. Note that we still treat the message in blocks but now the channel is a vector  $\mathbf{h} = \{h_1, \dots, h_L\}$ . The average SNR is  $\bar{\gamma} = \sigma_h^2 / \sigma_w^2$ .

### 3.4.2 Channel Estimation

By modeling the channel as a AR-1 process, we are able to use the Kalman filter to provide the linear MMSE channel estimate. We use periodic pilot symbols to

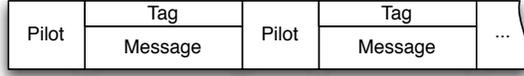


Figure 3.11: TDM Pilot Placement.

aid channel estimation but we use them more frequently because the channel is fast fading. We have  $T_p$  pilot symbols preceding every cluster of  $T_d$  data (i.e., message and tag) symbols and we let  $T = T_p + T_d$ . Thus pilots are inserted into  $\mathbf{x}$  such that  $\{x_k\}_{(k \bmod T < T_p)}$  are pilots and the rest are data. See Figure 3.11.

The channel estimation is slightly different depending on whether the tag presence is unknown or if it is assumed to be present. The presence is unknown, for example, by the unaware receiver, the aware receiver without the key, or the aware receiver who hasn't been able to verify it yet. However, once the intended receiver verifies the presence of the tag, it may use the tag as extra information to help estimate the channel.

### Tag Presence Unknown

The equations for channel state (3.65) and observation (3.66) are used to construct the filter. The filter trains itself to make increasingly accurate estimates while it is receiving the pilot symbols  $p_k$ . We have the following filter update equations during the training period ( $k \bmod T < T_p$ ) [30]:

$$[\text{Kalman gain}] K_k = \frac{(a^2 M_{k-1} + \sigma_u^2) p_k}{\sigma_w^2 + (a^2 M_{k-1} + \sigma_u^2) \sigma_p^2} \quad (3.68)$$

$$[\text{Estimate}] \hat{h}_k = a \hat{h}_{k-1} + K_k (y_k - a \hat{h}_{k-1} p_k) \quad (3.69)$$

$$[\text{MMSE}] M_k = (1 - K_k p_k) \cdot (a^2 M_{k-1} + \sigma_u^2). \quad (3.70)$$

When the training period is over, the filter estimates the channel based on the AR-1 model (3.65). The update equations during the data period ( $k \bmod T \geq T_p$ ) are:

$$[\text{Channel Estimate}] \hat{h}_k = a\hat{h}_{k-1}$$

$$[\text{MMSE}] M_k = a^2 M_{k-1} + \sigma_u^2$$

The channel estimate for the  $i^{\text{th}}$  block is the vector  $\hat{\mathbf{h}}_i$ .

### Tag Assumed Present

The aware receiver with the secret key can potentially obtain a better channel estimate than the unaware receiver. Recall that for authentication, our authentication tags must be known at the receiver. Therefore they may be used for channel estimation, in exactly the way as pilot symbols, provided that the tag is indeed present. The receiver that uses this information operates as follows. As soon as it can generate the estimated tag using (3.10), it uses  $\hat{\mathbf{t}}_i$  to adaptively track the channel during data symbol reception. Because the channel estimation does not change during the pilot symbol reception, equations (3.68) - (3.70) do not change.

When the data symbols are received however, the Kalman filter continues to update and track the signal by using the tag which it decides is present. Assuming that the estimated tag is present, we rewrite the observation

$$y_k = \rho^s h_k s_k + \rho^t h_k t_k + w_k \tag{3.71}$$

$$= \rho^t h_k t_k + v_k \tag{3.72}$$

Note that  $v_k \sim N(0, (\rho^s)^2 \sigma_h^2 + \sigma_w^2)$ . The update equations during the training

period ( $k \bmod T < T_p$ ) are [30]:

$$[\text{Kalman gain}] K_k = \frac{(a^2 M_{k-1} + \sigma_u^2) \rho^t t_k}{\sigma_v^2 + (a^2 M_{k-1} + \sigma_u^2) (\rho^t)^2} \quad (3.73)$$

$$[\text{Estimate}] \hat{h}_k = a \hat{h}_{k-1} + K_k (y_k - a \rho^t \hat{h}_{k-1} t_k) \quad (3.74)$$

$$[\text{MMSE}] M_k = (1 - \rho^t K_k t_k) \cdot (a^2 M_{k-1} + \sigma_u^2) \quad (3.75)$$

Comparing equations (3.73)-(3.75) with (3.68)-(3.70) reveals that  $\sigma_w^2$  is replaced with  $\sigma_v^2$  and  $p_k$  is replaced with  $\rho^t t_k$ . The channel estimate that assumes the tag is present for the  $i^{\text{th}}$  block is given by the vector  $\hat{\mathbf{h}}_i$ .

### 3.4.3 Message Recovery

#### Tag Presence Unknown

As before, the receiver uses its channel estimate  $\hat{\mathbf{h}}$  to estimate the message signal

$$x_k = \frac{\hat{h}_k^*}{|\hat{h}_k|^2} y_k \quad (3.76)$$

and uses equation (3.20) to recover the message symbols as before.

#### Tag Assumed Present

If the receiver decides that the tag is present, not only can it remove it prior to message estimation, it can also use the improved channel estimate  $\hat{\mathbf{h}}_i^+$ . The estimated message signal is then

$$x_k = \frac{1}{\rho^s} \left( \frac{(\hat{h}_k^+)^*}{|\hat{h}_k^+|^2} y_k - \rho^t t_k \right) \quad (3.77)$$

and uses equation (3.20) to recover the message symbols as before.

### 3.4.4 Authentication

The authentication process remains unchanged. Of course, the channel estimate used in the tag detection should not use the tag as pilot symbols, otherwise the reasoning is circular (testing the tag presence while assuming that it is there for channel estimation).

### 3.4.5 Example and Results

The following results are obtained when both tag and message are modulated using 4-QAM. The resultant constellation is the 4/16 hierarchical QAM constellation (Figure 3.2). Two pilot symbols precede every cluster of 8 message and tag symbols ( $T_p = 2$ ,  $T_d = 8$ ), and the channel is tracked using a Kalman filter. The Monte Carlo simulation parameters are summarized in Table 3.2.

A range of false alarm probabilities was simulated, and the pictured false alarm probability was chosen to give an example of a reasonable operating point. We note that as the false alarm probability becomes smaller, the power of the authentication test does not change much because the tails of the distribution under  $H_0$  (3.35) are very small. With suitably low false alarms, the authentication decisions may be trusted with high probability.

Figure 3.12 shows that as in the Rayleigh fading case, increasing message power improves the stealth of the system. It also shows the impact of message power upon frame outages and the importance of error correction codes in fast fading channels.

Since we authenticate only unmodified messages, in the following we assume that the message is received without error. Figure 3.13 shows that as in the block fading case, the message and tag power plays a significant role in the robustness of authentication. Again, we note that the benefit of using higher powers diminishes

Table 3.2: Simulation parameters for the single carrier, Gauss-Markov channel case

Channel Model	Gauss-Markov (AR-1)
Noise Model	AWGN
Channel Estimate Method	Kalman filtering
# Pilot Symbols	2 pilots every 8 data symbols
Frame Length	128 symbols
False Alarm Probability	$10^{-3}$
# Monte Carlo Samples	$2^{15}$

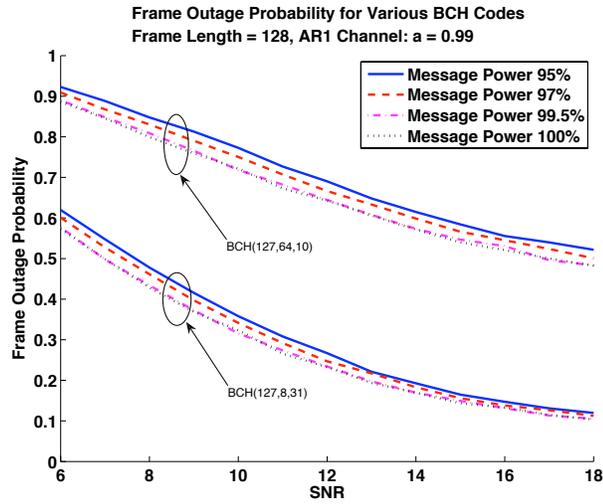


Figure 3.12: AR-1 Stealth: Frame outage probabilities for various BCH codes in the AR-1 channel with  $a = 0.99$ . Higher message power admits lower frame outage probability and hence better stealth.

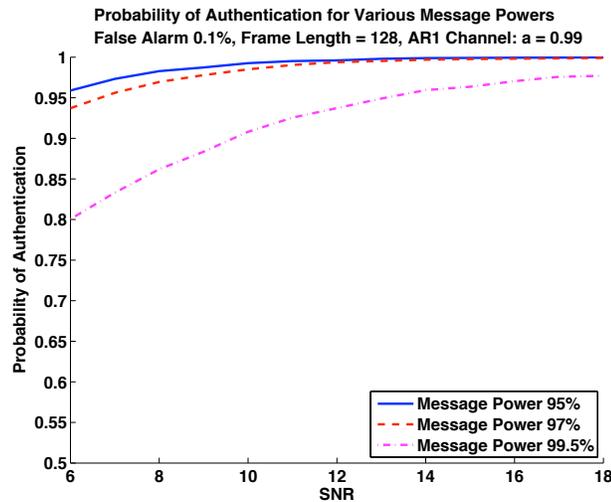


Figure 3.13: AR-1 Robustness: Frame authentication probabilities for various message powers in the AR-1 channel with  $a = 0.99$ . Lower message power admits higher authentication probability and hence better robustness.

at higher SNR.

Finally, Figure 3.14 shows that while the high BER induced by the channel is detrimental to robustness, it is favorable for security as it obscures the tag symbols very well. The equivocation of each 128 symbol = 256 bit tag is near the maximum (256) for reasonably high message powers as shown in the figure. The more uncertainty that is present in the channel (e.g., low SNR), the better the security and the less it depends on the message power.

### 3.5 Conclusion

We have presented a flexible framework for describing and analyzing a large family of physical layer authentication schemes that can be built over existing transmission systems. Authentication information is sent concurrently with data without requiring extra bandwidth or transmission power. With these constraints, en-

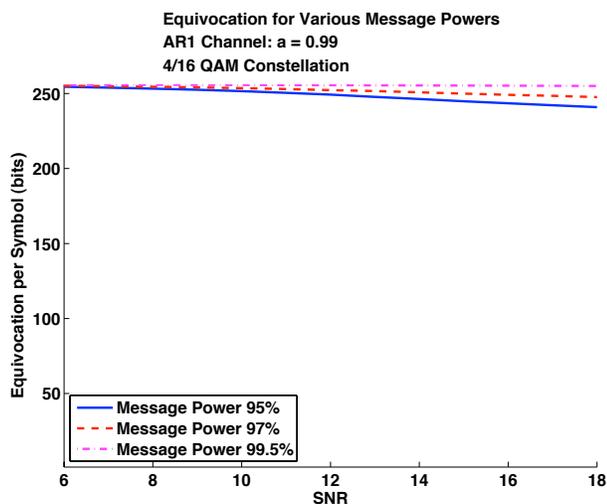


Figure 3.14: AR-1 Security: Equivocation of tag symbols for various message powers in the AR-1 channel with  $a = 0.99$ . Higher message power leads to higher equivocation and hence better security.

energy is allocated away from the data signal to the authentication signal, thereby increasing the probability of error of data recovery.

However, with a long enough authentication code word a useful authentication system can be achieved with very slight data degradation. Alternatively, it is also possible to place the authentication over multiple frames to further improve the system's robustness.

We have also presented the extension of the framework to fast-fading channels. Coding is essential in such channels to combat the high bit error rates so that authentication can proceed. At the same time, it is the high bit error rates that offer such good security properties so that the adversary gains very little information about the secret key.

## Chapter 4

### Multi-Carrier Authentication

#### 4.1 Overview of Contributions

In this chapter we extend the previous chapter's framework to describe authentication over multiple carriers. We expand the framework to handle unequal per-carrier power allocations and inter-carrier interference [31]. The added flexibility of having multiple carriers introduces new possibilities: varying tag placements over the carriers on a frame-by-frame basis, varying the density of the tags, and allocating unequal powers to each carrier [32].

We give a preview of the results as follows.

- We extend the framework of the previous chapter to allow the authentication to be transmitted over multiple carriers (Section 4.2).
- The number of unique carriers used to transmit the authentication tags is a major determining factor in the performance of the system. Spreading the authentication over more carriers improves stealth, robustness, and security (Sections 4.2.4 and 4.3). However, practical considerations such as hardware limits and quantization resolutions cap the number of carriers that can actually be used (Section 4.2.5).

- The errors introduced through a non-zero frequency offset and imperfect channel estimation degrade the performance of the authentication but may be overcome through careful selection of parameters: tag power, length, and density (Section 4.3).
- The availability of multiple carriers introduces additional degrees of freedom for hiding and securing the authentication. As a logical extension towards unequal power allocations, the authentication tags can exist on only a subset of the frame: that is, over specific carriers during specific times only. This further complicates the adversary's goal of defeating the authentication because she does not know where the tag is located (Section 4.3.3).
- When the adversary does not know where the authentication is located, it is more difficult for her to gain relevant information from her observations. Indeed, if she extracts data where none is present, she can effectively poison her cache of previously collected data (Section 4.3.3).
- When channel state information is available at the transmitter, the tag power allocation between carriers is another major factor in authentication performance. A simple strategy of proportional power allocation between message and tag is shown to yield a good compromise between the metrics (Section 4.4).

## 4.2 Framework

A note on notation: Bold face indicates matrices (e.g.  $\mathbf{A}$ ). Upper case indicates signals in the frequency domain (e.g.  $\mathbf{H}$ ). Lower case indicates in the time domain (e.g.  $\mathbf{h}$ ).

### 4.2.1 Signal Model

Suppose that Alice and Bob communicate using multiple carriers. In general, some carriers will be nulled out for spectral shaping purposes, but this does not have a significant impact on the authentication framework. Therefore we ignore the null carriers and assume that there are  $N > 1$  message carriers.

The signals are transmitted in frames represented by size  $N \times N^f$  matrices where  $N^f$  is the frame length. We assume the signals are i.i.d. and so we do not use time indices. Denote the transmitted signal by the random matrix  $\mathbf{X}$  with complex entries  $\{X(m, n)\}$  that have variance  $\sigma_x^2$ . We constrain the energy as given by its Frobenius norm

$$|\mathbf{X}|^2 = \text{Trace}(\mathbf{X}^H \mathbf{X}) \quad (4.1)$$

$$E|\mathbf{X}|^2 = NN^f \sigma_x^2 \quad (4.2)$$

First we consider **untagged** signals which are message-only (contain no authentication). The transmitted signal is

$$\mathbf{X} = \rho \mathbf{S} \quad (4.3)$$

where  $\rho$  is a  $N \times N$  diagonal scaling matrix and  $\mathbf{S}$  is a  $N \times N^f$  message matrix satisfying

$$E[S(m, n)] = 0 \quad (4.4)$$

$$E[|S(m, n)|^2] = \sigma_x^2 \quad (4.5)$$

$$\sum_{m,n} I(S(m, n)) = NN^f \quad (4.6)$$

where  $I(\cdot)$  is the indicator function. That is, the message symbols have zero mean and variance  $\sigma_x^2$ , and they occupy each of the  $NN^f$  symbol positions in the frame. The term  $\rho$  is used to allocate power among the carriers such that equations (4.2) and (4.3) are satisfied.

The **tagged** signals are formed by superimposing the authentication tag  $\mathbf{T}$  with the message  $\mathbf{S}$ :

$$\mathbf{X} = \rho^s \mathbf{S} + \rho^t \mathbf{T} \quad (4.7)$$

where  $\rho^s, \rho^t$  are  $N \times N$  diagonal scaling matrices and  $\mathbf{T}$  is a  $N \times N^f$  tag matrix satisfying

$$E[T(m, n)] = 0 \quad (4.8)$$

$$E[|T(m, n)|^2] = \sigma_x^2 I(T(m, n)) \quad (4.9)$$

$$\sum_{m,n} I(T(m, n)) = N^t N^f \quad (4.10)$$

Note that the tag symbols occupy  $N^t N^f$  out of a possible  $NN^f$  symbol positions with  $0 \leq N^t \leq N$ . When present, each tag symbol has zero mean and variance  $\sigma_x^2$ .  $N^t$  specifies the **spread** of the tag across the symbols; a large spread indicates that the tag energy is very spread out over many symbols while a small spread indicates that the tag energy is concentrated over only a few symbols. Figure 4.1a is an example of how the tag symbols may be scattered across carriers. Alternatively, the tag symbols may also occupy specific carriers as in Figure 4.1b. In this case,  $N^t$  is the number of carriers that are occupied by the authentication tag.

Denote the  $k^{th}$  row of a matrix by  $(\cdot)_k$ . For diagonal matrices such as  $\rho$  we slightly abuse the notation to write  $\rho_k \triangleq \rho(k, k)$ . The terms  $\rho^s, \rho^t$  are chosen to normalize the energy of tagged and untagged signals on each carrier:

$$E|\rho \mathbf{S}_k|^2 = E|\rho_k^s \mathbf{S}_k + \rho_k^t \mathbf{T}_k|^2 \quad (4.11)$$

where

We assume that the message and tag are uncorrelated

$$E[\text{Trace}(\mathbf{S}^H \mathbf{T})] = 0 \quad (4.12)$$

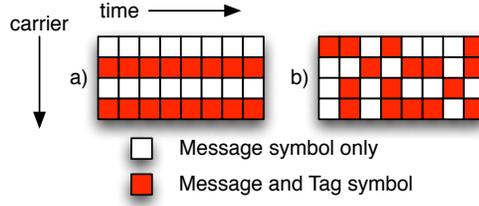


Figure 4.1: Example tag placements with  $N = 4$ ,  $N^f = 8$  and tag spread  $N^t = 2$ .

a) tag on specific carriers only, b) general tag placement.

so that the power constraint becomes

$$\rho_k^2 = (\rho_k^s)^2 + \frac{E|\mathbf{T}_k|^2}{E|\mathbf{S}_k|^2}(\rho_k^t)^2 \quad (4.13)$$

Note that  $|\rho^s|^2$  (resp.  $|\rho^t|^2$ ) is simply the overall percentage of power allocated to the message (resp. tag) symbols. The untagged signal is a special case of the tagged signal where  $\rho^s = \rho$  and  $\rho^t = \mathbf{0}$ . We thus use the more general formulation of equation (4.7) to represent both tagged and untagged signals.

Alice wants to send the message  $\mathbf{B}$  to Bob. They also share a secret key  $k \in \mathcal{K}$ , where  $|\mathcal{K}| = K$ . The messages and tags are generated as follows

$$\mathbf{S} = f_e(\mathbf{B}) \quad (4.14)$$

$$\mathbf{T} = g(\mathbf{B}, k) \quad (4.15)$$

where  $f_e(\cdot)$  is the message encoding function and  $g(\cdot)$  is the tag generation function. The encoding function  $f_e(\cdot)$  encapsulates any coding, modulation, or pulse shaping that may be used. The corresponding decoding function  $f_d(\cdot)$  is used at the receiver and satisfies

$$\mathbf{B} = f_d(f_e(\mathbf{B})) \quad (4.16)$$

for all possible inputs  $\mathbf{B}$  of  $f_e(\cdot)$ . For example, suppose that  $f_e(\cdot)$  applies a cyclic code (such as Reed-Solomon (RS) and Bose-Chaudhuri-Hocquenghem (BCH))

to the raw data  $\mathbf{B}$ . Then the corresponding decoder  $f_d(\cdot)$  can implement the Berlekamp-Massey algorithm [33] to efficiently decode the message.

The tag generating function  $g(\cdot)$  is assumed to be one-way, i.e., it is easy<sup>1</sup> to calculate  $\mathbf{T}$  given  $\mathbf{B}$  and  $k$ , but hard to find  $k$  given  $\mathbf{T}$  and  $\mathbf{B}$ . Further, it is collision resistant so that it is hard to find  $\mathbf{X} \neq \mathbf{Y}$  such that  $g(\mathbf{X}, k) = g(\mathbf{Y}, k)$ . This property ensures resistance against the substitution attack (Section 4.3.3).

The transmitted (time domain) signal  $\mathbf{x}$  is obtained by taking the IDFT of  $\mathbf{X}$

$$\mathbf{x} = \text{IDFT}[\mathbf{X}] \quad (4.17)$$

$$= \mathbf{F}^H \mathbf{X} \quad (4.18)$$

where  $\mathbf{F}$  is the unitary  $N \times N$  FFT matrix with entries

$$F(m, n) = \frac{1}{\sqrt{N}} \exp(-j2\pi mn/N) \quad (4.19)$$

and  $0 \leq m, n \leq N - 1$ .

## 4.2.2 Channel Model and Estimation

We assume a block fading multipath channel. The channel  $\vec{\mathbf{h}}$  is modeled as a delay line with equally spaced taps

$$\vec{h}(m) = \sum_{l=0}^{L-1} \alpha(l) \delta(m-l) \quad (4.20)$$

where  $\alpha(\cdot)$  are i.i.d. complex zero-mean Gaussian random variables with variance  $N/L$ .

The frequency response of the channel is

$$\mathbf{H} = \text{diag}(\mathbf{F}\vec{\mathbf{h}}) \quad (4.21)$$

---

<sup>1</sup>The concept of *hard* and *easy* calculations is characterized by their feasibility. Hard calculations are infeasible to compute given constraints on computational resources, while easy calculations are feasible to compute under the same constraints.

where  $\vec{\mathbf{h}}$  is zero-padded to  $N$ , the length of the FFT (equation (4.19)). Note that the frequency response per carrier has unit expected variance ( $\sigma_h^2 = 1$ ). We use the convention that  $\text{diag}(\mathbf{X})$  returns the main diagonal of (matrix)  $\mathbf{X}$  as a column vector while  $\text{diag}(\vec{\mathbf{x}})$  returns the diagonal matrix with (vector)  $\vec{\mathbf{x}}$  on the main diagonal.

Cyclic prefixes with length  $N^g \geq L$  are inserted at the transmitter and removed at the receiver to eliminate intersymbol interference (ISI) and to make the convolution circular over the frame. Thus the convolutive channel may be written as a circulant matrix  $\tilde{\mathbf{h}}$  with entries

$$\tilde{h}(m, n) = h((m - n) \bmod N) \quad (4.22)$$

After stripping the cyclic prefix, the receiver has the observation

$$\mathbf{y} = \tilde{\mathbf{h}}\mathbf{x} + \mathbf{w} \quad (4.23)$$

where  $\mathbf{w}$  is a  $N \times N^f$  matrix with AWGN entries with variance  $\sigma_w^2$ . The SNR is  $\gamma = \sigma_x^2 \sigma_h^2 / \sigma_w^2$ .

The receiver takes the FFT of the observation  $\mathbf{y}$  to work in the frequency domain. In general, the oscillators of the transmitter and receiver have a non-zero frequency offset  $\varepsilon$  and timing error  $\theta$ . Synchronization may be done by taking advantage of the cyclic prefix [34] or an inserted preamble [35]. The frequency domain signal at the receiver is

$$\mathbf{Y}^{\varepsilon, \theta} = e^{-j\theta} \mathbf{Y}^\varepsilon \quad (4.24)$$

$$\text{where } \mathbf{Y}^\varepsilon = \text{DFT}[\mathbf{y}] \quad (4.25)$$

$$= \mathbf{F}\mathbf{E}\mathbf{y} \quad (4.26)$$

$$= \mathbf{F}\mathbf{E}\tilde{\mathbf{h}}\mathbf{F}^H \mathbf{X} + \mathbf{W} \quad (4.27)$$

$$= \mathbf{H}^\varepsilon \mathbf{X} + \mathbf{Z}^\varepsilon + \mathbf{W} \quad (4.28)$$

where

$$\mathbf{E} = \text{diag}(\exp(j2\pi\varepsilon m/N)) \quad (4.29)$$

$$\mathbf{H}^\varepsilon = \text{diag}(\text{diag}[\mathbf{F}\mathbf{E}\tilde{\mathbf{h}}\mathbf{F}^H]) \quad (4.30)$$

$$H^\varepsilon(k) = H(k)e^{j\pi\varepsilon(N-1)/N} \frac{\sin(\pi\varepsilon)}{N \sin(\pi\varepsilon/N)} \quad (4.31)$$

$$\mathbf{Z}^\varepsilon = (\mathbf{F}\mathbf{E}\tilde{\mathbf{h}}\mathbf{F}^H - \mathbf{H}^\varepsilon)\mathbf{X} \quad (4.32)$$

$$Z^\varepsilon(k) = \sum_{\substack{l=0 \\ l \neq k}}^{N-1} H(l)X(l)e^{j\pi\varepsilon(N-1)/N} e^{j\pi(l-k)/N} \\ * \frac{\sin(\pi\varepsilon)}{N \sin(\pi(l-k+\varepsilon)/N)} \quad (4.33)$$

In the perfect synchronization case we have  $\varepsilon = 0$  and  $\mathbf{E} = \mathbf{I}$ . Since  $\tilde{\mathbf{h}}$  is circulant,  $\mathbf{F}\tilde{\mathbf{h}}\mathbf{F}^H$  is diagonal and therefore  $\mathbf{Z} = 0$ . Therefore there is no intercarrier interference (ICI) when the frequency offset is zero.

We use pilot symbol assisted modulation (PSAM) to estimate the channel. The proposed authentication scheme only modifies the message symbols; no control symbols such as pilots are perturbed. The pilot symbols and subsequent channel estimates are transparent to our method, but we characterize the estimation noise in the following.

The channel is held constant over the entire message by the slow fading assumption. The least squares estimate of the channel is the column vector

$$\vec{H}^{ls}(k) = \frac{Y_p(k, m)}{P(k, m)} \quad (4.34)$$

where  $\mathbf{P}$  are the known pilot symbols and  $\mathbf{Y}_p$  are the corresponding observations. This zero-forcing estimate performs poorly with severely attenuated carriers which is a hallmark of fading channels, and thus we use the LMMSE estimate [36]

$$\vec{\mathbf{H}}^{lmmse} = \mathbf{R}_{\mathbf{H}\mathbf{H}}(\mathbf{R}_{\mathbf{H}\mathbf{H}} + \sigma_w^2(\mathbf{P}\mathbf{P}^H)^{-1})^{-1}\hat{\mathbf{H}}^{ls} \quad (4.35)$$

The LMMSE estimate may be simplified by substituting  $E[(\mathbf{P}\mathbf{P}^H)^{-1}]$  for  $(\mathbf{P}\mathbf{P}^H)^{-1}$  in equation (4.35) [37]. The simplified estimator may thus be written as the diagonal matrix

$$\hat{\mathbf{H}} = \text{diag}(\mathbf{R}\vec{\mathbf{H}}^{ls}) \quad (4.36)$$

where

$$\mathbf{R} = \mathbf{R}_{\mathbf{H}\mathbf{H}} \left( \mathbf{R}_{\mathbf{H}\mathbf{H}} + \frac{\beta}{\gamma} \mathbf{I} \right)^{-1} \quad (4.37)$$

$$\beta = E[\mathbf{P}\mathbf{P}^H]E[(\mathbf{P}\mathbf{P}^H)^{-1}] \quad (4.38)$$

$\beta$  is a constant that depends on the constellation of  $\mathbf{P}$ . Note that when  $\mathbf{P}$  is constant modulus (e.g., 4QAM), the simplified estimator gives the same channel estimate, i.e.,  $\hat{\mathbf{H}} = \text{diag}(\vec{\mathbf{H}}^{mmse})$ .

The channel autocorrelation matrix is determined by the number of channel taps  $L$ :

$$\mathbf{R}_{\mathbf{H}\mathbf{H}} = E[\mathbf{H}\mathbf{H}^H] \quad (4.39)$$

$$\mathbf{R}_{\mathbf{H}\mathbf{H}}(m, n) = \sum_{l=0}^{L-1} \exp(-j2\pi l(m-n)/N) \quad (4.40)$$

for  $0 \leq m, n \leq N-1$ .

The channel estimate may be viewed as a noisy observation of the true channel

$$\hat{\mathbf{H}} = \mathbf{H}^\varepsilon + \eta \quad (4.41)$$

where  $\eta$  is the channel estimation error. Note that  $\eta$  is a diagonal matrix since  $\mathbf{H}$  and  $\mathbf{H}^\varepsilon$  are both diagonal.

$$\eta = \hat{\mathbf{H}} - \mathbf{H}^\varepsilon \quad (4.42)$$

$$= \mathbf{R} \left( \mathbf{H}^\varepsilon + \frac{\mathbf{Z}^\varepsilon + \mathbf{W}}{\mathbf{P}} \right) - \mathbf{H}^\varepsilon \quad (4.43)$$

$$= (\mathbf{R} - \mathbf{I})\mathbf{H}^\varepsilon - \mathbf{R} \frac{\mathbf{Z}^\varepsilon + \mathbf{W}}{\mathbf{P}} \quad (4.44)$$

The MSE is

$$\sigma_\eta^2 = \frac{1}{N} E[\text{tr}(\eta^H \eta)] \quad (4.45)$$

The error  $\eta$  can be approximated by a zero-mean Gaussian random variables with variance  $\sigma_\eta^2$ .

### 4.2.3 Message Recovery

Using the channel estimate, the receiver estimates the message signal as

$$\hat{\mathbf{S}} = \hat{\mathbf{H}}^{-1} \mathbf{Y} \quad (4.46)$$

$$= \hat{\mathbf{H}}^{-1} (\mathbf{H}^\varepsilon \mathbf{X} + \mathbf{Z}^\varepsilon + \mathbf{W}) \quad (4.47)$$

$$= \mathbf{X} + \hat{\mathbf{H}}^{-1} (-\eta \mathbf{X} + \mathbf{Z}^\varepsilon + \mathbf{W}) \quad (4.48)$$

$$= \mathbf{X} + \hat{\mathbf{H}}^{-1} \tilde{\mathbf{W}} \quad (4.49)$$

The estimated message is

$$\hat{\mathbf{B}} = f_d(\hat{\mathbf{S}}) \quad (4.50)$$

where  $f_d(\cdot)$  is the decoding function corresponding to the encoder  $f_e(\cdot)$  from equation (4.16).

Now let us consider the uncoded bit error probability of the symbols. For analysis, we assume that the message and tag symbols are modulated using QAM. For example, when the message and tag symbols are both modulated with 4-QAM, the tag constellation is superimposed on each message symbol to form the constellation shown in Figure 3.2. In the literature this is called the 4/16 hierarchical QAM constellation [27]. Note that the constellation has 16 symbols; each symbols signals both message (which quadrant) and tag (which point in the quadrant) information.

To calculate the BER for hierarchical QAM constellations, we must know the distance between the symbols as well as the noise power (Figure 3.2).  $2d_1$  is the minimum distance between any two (fictitious) message points,  $2d_2$  is the minimum distance between any two tag points within the same quadrant, and  $2d'_1$  is the minimum distance between points in adjacent quadrants. We first give the exact BER expressions for unit-energy 4/16 constellations and then apply them to the authentication signals. In general, any number of QAM constellations may be superimposed on each other and the corresponding closed form BER expressions are given in [27].

The BER of the message bit [27] is

$$p^s = \frac{1}{2} \left( \frac{1}{2} \operatorname{erfc} \frac{d'_1}{\sqrt{N_0}} + \frac{1}{2} \operatorname{erfc} \frac{d'_1 + 2d_2}{\sqrt{N_0}} \right) \quad (4.51)$$

$$= \frac{1}{4} (\Psi(1, 0) + \Psi(1, 2)) \quad (4.52)$$

where  $N_0$  is the noise power, while the BER of the tag bit is

$$p^t = \frac{1}{2} \left( \operatorname{erfc} \frac{d_2}{\sqrt{N_0}} + \frac{1}{2} \operatorname{erfc} \frac{2d'_1 + d_2}{\sqrt{N_0}} - \frac{1}{2} \operatorname{erfc} \frac{2d'_1 + 3d_2}{\sqrt{N_0}} \right) \quad (4.53)$$

$$= \frac{1}{4} (2\Psi(0, 1) + \Psi(2, 1) + \Psi(2, 3)) \quad (4.54)$$

The helper function  $\Psi(\cdot)$  is used in the BER calculations above, and depends on the channel distribution. For the Rayleigh channel with unit power constellations it is

$$\Psi(a, b) = 1 - \sqrt{\frac{(ad'_1 + bd_2)^2 \gamma}{1 + (ad'_1 + bd_2)^2 \gamma}} \quad (4.55)$$

The noise power  $N_0$  is the average noise power for a unit variance channel. For perfect channel information,  $1/\gamma$ , while for MMSE estimation it is  $1/\gamma^{lmmse} > 1/\gamma$ .

We now evaluate the parameters for the BER expressions in equations (4.51) and (4.53) for two cases: when the message symbol is tagged and when it is not.

**Case 1:** Tagged message symbol

For a symbol on carrier  $k$  that contains both message and tag, we have that the message symbol is scaled by  $\rho_k^s$  and the tag symbol is scaled by  $\rho_k^t$ . Since the symbols are each unit variance, the effective SNR is  $[(\rho_k^s)^2 + (\rho_k^t)^2]/\sigma_w^2$ . In order to use Equations (4.51) and (4.53), the constellation needs to be unit energy. Thus we scale

$$\tilde{\rho}_k^s = \rho_k^s / \sqrt{(\rho_k^s)^2 + (\rho_k^t)^2} \quad (4.56)$$

$$\tilde{\rho}_k^t = \rho_k^t / \sqrt{(\rho_k^s)^2 + (\rho_k^t)^2} \quad (4.57)$$

$$N_0 = \sigma_w^2 / [(\rho_k^s)^2 + (\rho_k^t)^2] \quad (4.58)$$

$$d_1 = \tilde{\rho}_k^s \sqrt{2} \quad (4.59)$$

$$d_2 = \tilde{\rho}_k^t \sqrt{2} \quad (4.60)$$

$$d'_1 = d_1 - d_2 \geq 0 \quad (4.61)$$

We calculate the BER of the message and tag bits by using these values in equations (4.51) and (4.53) respectively. Note that equations (4.59)-(4.61) require that  $\tilde{\rho}_k^s > \tilde{\rho}_k^t$  which implies  $\rho_k^s > \rho_k^t$ . Otherwise, the decision boundaries will overlap.

**Case 2:** Untagged message symbol

When a message symbol on carrier  $k$  stands alone without any superimposed tag, it uses the 4-QAM constellation. Note that it is still scaled by the term  $\rho_k^s$ , and thus the effective SNR is  $(\rho_k^s)^2/\sigma_w^2$ . Once again we scale in order to make the

constellation unit energy:

$$\tilde{\rho}_k^s = 1 \quad (4.62)$$

$$\tilde{\rho}_k^t = 0 \quad (4.63)$$

$$N_0 = \sigma_w^2 / (\rho_k^s)^2 \quad (4.64)$$

and calculate the following parameters:

$$d_1 = \tilde{\rho}_k^s \sqrt{2} \quad (4.65)$$

$$d_2 = 0 \quad (4.66)$$

$$d'_1 = d_1 \quad (4.67)$$

The overall BER of the system is calculated by combining the results of cases 1 and 2. Recall that the expected number of tagged message symbols on carrier  $k$  is  $E|\mathbf{T}_k|^2$ . By using  $p_{k,1}^s, p_{k,2}^s$  to denote the message BER for the cases 1 and 2 respectively and similarly  $p_{k,1}^t, p_{k,2}^t$  for the tag BER, the overall message and tag BERs are

$$p^s = \sum_k \frac{E|\mathbf{T}_k|^2}{Nf} p_{k,1}^s + \left(1 - \frac{E|\mathbf{T}_k|^2}{Nf}\right) p_{k,2}^s \quad (4.68)$$

$$p^t = \sum_k \frac{E|\mathbf{T}_k|^2}{Nf} p_{k,1}^t + \left(1 - \frac{E|\mathbf{T}_k|^2}{Nf}\right) p_{k,2}^t \quad (4.69)$$

In the above we have assumed that each carrier transmits the same number of bits on each carrier; otherwise, the BER calculation would have to be weighted differently for each carrier. This is potentially the case when channel state information is available at the transmitter (Section (4.4)).

#### 4.2.4 Tag Recovery

With his estimate of the data  $\hat{\mathbf{B}}$ , Bob uses  $g(\cdot)$  from equation (4.15) to reconstruct the estimated tag:

$$\hat{\mathbf{T}} = g(\hat{\mathbf{B}}, k) \quad (4.70)$$

Bob uses matched filtering to detect it in his observation  $\mathbf{Y}$ . He calculates the residual  $\mathbf{R}$  by removing the message and then correlates it with the estimated tag to obtain the test statistic  $\tau$ .

$$\mathbf{R} = \mathbf{Y} - \rho^s \hat{\mathbf{H}} f_e(\hat{\mathbf{B}}) \quad (4.71)$$

$$\tau = \Re(\text{tr}((\rho^t \hat{\mathbf{H}} \hat{\mathbf{T}})^H \mathbf{R})) \quad (4.72)$$

The receiver decides between the hypotheses

$$H_0 : \quad \hat{\mathbf{T}} \text{ is not present in } \mathbf{R} \quad (4.73)$$

$$H_1 : \quad \hat{\mathbf{T}} \text{ is present in } \mathbf{R} \quad (4.74)$$

The authentication decision  $\delta$  is made according to the threshold test

$$\delta = \begin{cases} 0 & \tau < \tau^0 \\ 1 & \tau \geq \tau^0 \end{cases} \quad (4.75)$$

where the threshold  $\tau^0$  is determined for a false alarm probability  $\alpha$  according to the distribution of  $(\tau|H_0)$ . As in the single carrier case, the authentication is low-complexity because the required tag generation and correlation are simple operations.

We now turn our attention to the performance of the authentication test. We first consider the false alarm probability and how the threshold is calculated. Then we consider the power of the test, which in this case is the probability of accepting authentic messages.

## False Alarm Probability

In order to limit the false alarm probability  $\alpha$ , we calculate the threshold  $\tau^0$  such that  $P(\tau > \tau^0 | H_0) \leq \alpha$ . There are two cases where a false alarm can occur: when the observation contains no tag at all or when the observation contains an incorrect tag.

We assume that the message is recovered without error because that is when authentication is useful. That is,  $\hat{\mathbf{B}} = \mathbf{B}$ . Consider the structure of the residual  $\mathbf{R}$

$$\mathbf{R} = \mathbf{H}^\varepsilon \mathbf{X} + \mathbf{Z}^\varepsilon + \mathbf{W} - \rho^s \hat{\mathbf{H}} \mathbf{S} \quad (4.76)$$

$$= (\mathbf{H}^\varepsilon + \eta - \eta) \mathbf{X} + \mathbf{Z}^\varepsilon + \mathbf{W} - \rho^s \hat{\mathbf{H}} \mathbf{S} \quad (4.77)$$

$$= \hat{\mathbf{H}}(\mathbf{X} - \rho^s \mathbf{S}) - \eta \mathbf{X} + \mathbf{Z}^\varepsilon + \mathbf{W} \quad (4.78)$$

$$= \hat{\mathbf{H}}(\mathbf{X} - \rho^s \mathbf{S}) + \tilde{\mathbf{W}} \quad (4.79)$$

and  $\sigma_w^2 = \eta^s \sigma_x^2 + \sigma_z^2 + \sigma_w^2$ .

We consider the two cases where a false alarm can occur: when the signal contains no tag or when it contains an invalid tag. The cases differ in the content of the signal  $\mathbf{X}$ .

**Case 1:** the transmitted signal does not contain any tag (i.e.,  $\mathbf{X} = \rho \mathbf{S}$ ). Then

$$\tau | H_0 = \Re(\text{tr}((\rho^t \hat{\mathbf{H}} \hat{\mathbf{T}})^H \mathbf{R})) \quad (4.80)$$

$$= \Re\left(\text{tr}((\rho^t \hat{\mathbf{H}} \hat{\mathbf{T}})^H ((\rho - \rho^s) \hat{\mathbf{H}} \mathbf{S} + \tilde{\mathbf{W}}))\right) \quad (4.81)$$

$$= \Re\left(\text{tr}(\rho^t (\rho - \rho^s) (\hat{\mathbf{H}}^H \hat{\mathbf{H}}) \hat{\mathbf{T}}^H \mathbf{S})\right) + v \quad (4.82)$$

where  $v$  is a real Gaussian variable with zero mean and variance  $\sigma_v^2 = |\rho^t|^2 |\hat{\mathbf{H}}|^2 |\hat{\mathbf{T}}|^2 \sigma_w^2$ .

Consider the term  $\Re(\text{tr}(\hat{\mathbf{T}}^H \mathbf{S}))$ . It is a sum of  $N^t N^f$  i.i.d. variables and is well approximated by a Gaussian distribution when  $N^t N^f$  is large (central limit theorem). The mean is zero (4.12) and the variance depends on the symbol

constellations. For example, if the message and tag are composed of QPSK symbols, the variance of  $\Re(\text{tr}(\hat{\mathbf{T}}^H \mathbf{S}))$  is  $\sigma_{ts}^2 = N^t N^f * \frac{1}{4}(1 + 1 + 0 + 0) = \frac{N^t N^f}{2}$ . Thus the test statistic  $\tau$  is Gaussian with zero mean and variance

$$\sigma_\tau^2 = |\rho^t|^2 |\rho - \rho^s|^2 |\hat{\mathbf{H}}|^4 \sigma_{ts}^2 + \sigma_v^2 \quad (4.83)$$

**Case 2:** the transmitted signal contains a tag different from estimated tag (i.e.,  $\mathbf{T} \neq \hat{\mathbf{T}}$ ). Then

$$\tau|H_0 = \Re(\text{tr}((\rho^t \hat{\mathbf{H}} \hat{\mathbf{T}})^H \mathbf{R})) \quad (4.84)$$

$$= \Re(\text{tr}((\rho^t \hat{\mathbf{H}} \hat{\mathbf{T}})^H (\rho^t \hat{\mathbf{H}} \mathbf{T} + \tilde{\mathbf{W}})) \quad (4.85)$$

$$= \Re(\text{tr}((\rho^t)^2 (\hat{\mathbf{H}}^H \hat{\mathbf{H}}) \hat{\mathbf{T}}^H \mathbf{T})) + v \quad (4.86)$$

where  $v$  is defined as in case 1.

Consider the term  $\Re(\text{tr}(\hat{\mathbf{T}}^H \mathbf{T}))$ . It is a sum of  $N^t N^f$  i.i.d. variables and is well approximated by a Gaussian distribution when  $N^t N^f$  is large (central limit theorem). The mean is zero (4.12) and the variance depends on the symbol constellations. For example, if the tag is composed of QPSK symbols, the variance of  $\Re(\text{tr}(\hat{\mathbf{T}}^H \mathbf{S}))$  is  $\sigma_{ts}^2 = N^t N^f * \frac{1}{4}(1 + 1 + 0 + 0) = \frac{N^t N^f}{2}$ . Thus the test statistic  $\tau$  is Gaussian with zero mean and variance

$$\sigma_\tau^2 = |\rho^t|^4 |\hat{\mathbf{H}}|^4 \sigma_{ts}^2 + \sigma_v^2 \quad (4.87)$$

---

Without priors that indicate which case is applicable, the threshold is calculated based on the worst case distribution from either case 1 or 2. Since both are zero mean Gaussian distributions, the worst case has the larger variance  $\tilde{\sigma}_\tau^2$  from equations (4.83) and (4.87).

$$\tau^0 = \arg \min_{\tau} \Phi(\tau/\tilde{\sigma}_\tau) \geq 1 - \alpha \quad (4.88)$$

where  $\Phi(\cdot)$  is the standard Gaussian cumulative distribution function.

## Detection Probability

When Bob generates the correct tag ( $\hat{\mathbf{T}} = \mathbf{T}$ ), the test statistic is

$$\tau|_{H_1} = \Re(\text{tr}((\rho^t \hat{\mathbf{H}} \hat{\mathbf{T}})^H \mathbf{R})) \quad (4.89)$$

$$= \Re(\text{tr}((\rho^t)^2 (\hat{\mathbf{H}}^H \hat{\mathbf{H}}) \mathbf{T}^H \mathbf{T})) + v \quad (4.90)$$

$$= |\rho^t \hat{\mathbf{H}} \mathbf{T}|^2 + v \quad (4.91)$$

The probability of detection is

$$P^a(\gamma) = 1 - \Phi\left(\frac{\tau^0 - |\rho^t \hat{\mathbf{H}} \mathbf{T}|^2}{\sigma_v}\right) \quad (4.92)$$

As discussed in Chapter 3, the detection probability may be raised by increasing transmission power, extending the detection over multiple frames, or by coding the message.

When multiple carriers are available to Alice and Bob, one question that arises is how many carriers to use for authentication. A related question is how the authentication tag should be placed: over specific carriers or randomly over all carriers. We now turn our attention to how the usage of multiple carriers influences the authentication probability.

Recall that there are  $N$  carriers and  $N^t N^f$  tag symbols per frame. With  $N^f$  symbols per carrier, this means that the tag symbols will occupy between  $\lceil N^t \rceil$  and  $\min(N^t N^f, N)$  carriers.

Suppose that the tag occupies  $n$  carriers. Assume that the tag symbols are uniformly distributed among the  $n$  carriers. Recall from equation (4.91) that  $|\rho^t \hat{\mathbf{H}} \mathbf{T}|^2$  is the mean of the test statistic when the valid authentication tag is present. For clarity of discussion we assume that  $\rho^t$  scales each tagged carrier equally and that  $\hat{\mathbf{H}} = \mathbf{H}$  so we consider  $|\mathbf{H} \mathbf{T}|^2$  only. Since  $\mathbf{H}$  is diagonal with

unit variance entries (Section 4.2.2),

$$Q = |\mathbf{HT}|^2 \quad (4.93)$$

$$= \text{tr}(\mathbf{T}^H(\mathbf{H}^H\mathbf{H})\mathbf{T}) \quad (4.94)$$

$$= \text{tr}((\mathbf{T}^H\mathbf{T})(\mathbf{H}^H\mathbf{H})) \quad (4.95)$$

$$= \frac{N^t N^f}{n} \sum_k^n |H_k|^2 \quad (4.96)$$

$$= \frac{N^t N^f}{2n} \sum_j^{2n} Z_j^2 \quad (4.97)$$

where  $Z_j$  is a zero mean, unit variance Gaussian random variable. Thus  $Q$  is a (scaled)  $\chi^2$  distribution with  $2n$  degrees of freedom. We characterize the distribution by the following:

$$E[Q] = \mu_1 = N^t N^f \quad (4.98)$$

$$\sigma_Q^2 = \mu_2 = \frac{1}{n}(N^t N^f)^2 \quad (4.99)$$

$$\gamma_2 = \mu_4/\sigma_Q^4 - 3 \quad (4.100)$$

where  $\gamma_2$  is the kurtosis and  $\mu_4$  is the fourth central moment of  $Q$ . Increasing  $n$  decreases the both the variance and the kurtosis. This indicates the tightness of the distribution about the mean and the heaviness of the tails (larger kurtosis = heavier tail). Figure 4.2 shows the distribution of  $\text{tr}(Q)$  for various  $n$ .

Therefore, to improve the detection of the correct authentication tag, it is beneficial to spread the tag symbols over as many independent carriers as possible. That is, for the same spread factor  $N^t$ , it is better to scatter the tag over many carriers rather than restricting its placement to exactly  $N^t$  carriers (see Figure 4.1).

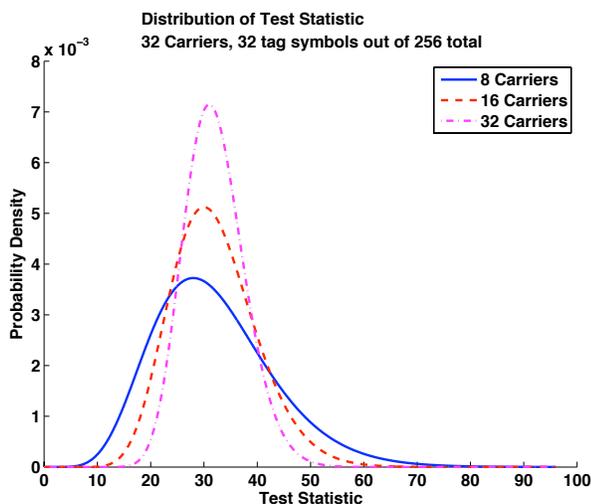


Figure 4.2: Distribution of test statistic when  $N = 32$ ,  $N^t = 8$ ,  $N^f = 4$ . Kurtosis values are  $0.69(n = 8)$ ,  $0.37(n = 16)$ ,  $0.19(n = 32)$ . Increasing  $n$  tightens the distribution about its mean, improving tag detection performance of the receiver.

#### 4.2.5 Practical Considerations

In the previous sections we have seen that the density of the tag  $N_t/N$  plays a major role in the performance of the system. However, as we see in this section, there are constraints on the number of carriers. We will first see why there is a minimum density requirement in order to maintain the stealth and security of the system. Then, we will examine the limitations of real hardware and the upper limit on the number of carriers that can be used for the authentication.

##### Lower Limit of $N_t$

The normalized power allocation in equation (4.13) limits the total powers of the tagged carrier to be the same as the untagged carrier. Note, however, that there is a fundamental difference between this allocation and that of the single-carrier case: a high message power  $(\rho_k^s)^2$  does not guarantee a low tag power  $(\rho_k^t)^2$

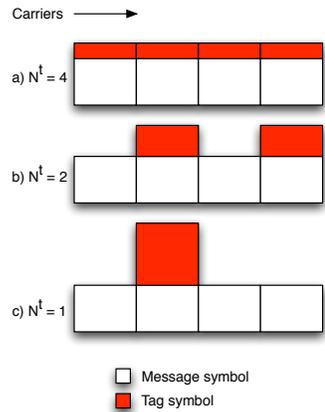


Figure 4.3: Each tag symbol becomes less powerful as the tag is spread out over more symbols.

anymore.

Consider the power allocation for carrier  $k$  in equation (4.13). Noting the dependence on the ratio  $\frac{E|\mathbf{T}_k|^2}{E|\mathbf{S}_k|^2} \leq 1$ , it follows that whenever the ratio becomes smaller, there are less tags on the carrier but the power of each tag symbol becomes greater.

For example, suppose that there are  $N = 4$  carriers and let  $N^f = 1$  so that there are exactly  $N$  message symbols and  $N^t$  tag symbols. Normalize the powers so that the average power is 1. If each message symbol is transmitted at 99% power, then each tag symbol is transmitted at  $\frac{N}{N^t} * 1\%$  power. Thus we have that when  $N^t = \{4, 2, 1\}$  the tag powers are respectively  $\{1, 2, 4\}\%$  respectively. The basic idea is conveyed in Figure 4.3.

In medium to high SNR regimes, the tagged symbols at this power are particularly noticeable. The observed distribution will become heavy tailed with such large perturbations, and it becomes easier for the adversary to 1) determine that authentication is being passed and 2) extract information about the secret key.

In order to maintain stealth and security, it is therefore necessary to impose

a lower limit on how many symbols are used to spread the authentication tag. This lower limit depends on the power allocation  $\rho^s$  and is relative the number of carriers  $N$  since we are concerned about the ratio of message to tag symbols. When the tag power is low, it is possible to concentrate the authentication over a few tag symbols without risking detection.

### Upper Limit of $N_t$

While theoretical performance forces us away from concentrating the authentication in only a few tag symbols, physical limitations can prevent us from using all possible symbol positions given the tag power available. For example, spreading the authentication over each message symbol may allocate infinitesimally little power to the tag symbols, which is not possible in real devices.

A digital transceiver has analog-to-digital (A/D) and digital-to-analog (D/A) converters connected to amplifiers. The perturbation that can be physically realized is dependent on the resolution of the D/A converters which are specified in bits. For example, a 10-bit D/A can generate distinct signals when the tag power is  $(\rho_k^t)^2 > 2^{-10}$ , or about 0.1%. Beyond this lower limit for the tag power, the signals will be identical regardless of tag content. That is, the tag will be completely obscured by quantization noise. In the same vein, the receiver's D/A should have sufficient resolution to distinguish between small perturbations.

Amplifiers have a region of operation that, when saturate when exceeded and hard-limit the output signal. This distortion is very undesirable so the input of the amplifier is scaled to so that, with high probability, it remains within the operation region. In OFDM systems, a serious problem is the high peak to average power ratio (PAPR). The PAPR compares the highest power of the signal with the RMS signal, and thus defines the dynamic range of the D/A. This means

that the D/A must quantize a wider range with the same number of bits, which further reduces the resolution of the input signal. PAPR reduction techniques are able to mitigate the effects to about 6 dB.

## 4.3 Metric Evaluation

We now discuss the desirable properties of the authentication system: stealth, robustness, and security. We will qualitatively and quantitatively give heuristics for system design.

In the simulations, we assume  $N = 32$  carriers. Each frame consists of  $N^f = 4$  OFDM symbols, for a total of  $32 * 4 = 128$  message symbols. The Monte Carlo simulation parameters are summarized in Table 4.1.

A range of false alarm probabilities was simulated, and the pictured false alarm probability was chosen to give an example of a reasonable operating point. We note that as the false alarm probability becomes smaller, the power of the authentication test does not change much because the tails of the distribution under  $H_0$  (4.73) are very small. With suitably low false alarms, the authentication decisions may be trusted with high probability.

### 4.3.1 Stealth

The stealth of the system is measured by the inability of the adversary (Eve) to distinguish between tagged and untagged signals. We discuss the presence and the impact of the authentication in turn.

Table 4.1: Simulation parameters for the multi-carrier, no CSI case

Channel Model	Rayleigh block fading
Noise Model	AWGN
Channel State Information?	No
# Carriers	32 (4 taps)
Channel Estimate Method	MMSE
# Pilot Symbols	1 OFDM symbol per frame
Frame Length	4 OFDM symbols
False Alarm Probability	$10^{-7}$
# Monte Carlo Samples	$2^{14}$

## Presence

In the previous chapter, we have seen how the presence of the tag is difficult to detect when the power is low. In the multicarrier case we consider how the detection depends on the spread of the authentication tag. Note that spreading the authentication over more symbols decreases the amount of perturbation per symbol. Conversely, concentrating the authentication over fewer symbols increases the perturbation per symbol. Thus it is not intuitively obvious whether it is better to hide the tag over a few high-powered bursts or over many low-powered perturbations.

We fix the message power while varying the ratio of tag symbols and consider the resulting Kullback-Leibler distance between tagged and untagged signals. Figure 4.4 shows that spreading the authentication over more symbols decreases the KL distance, and hence improves the stealth.

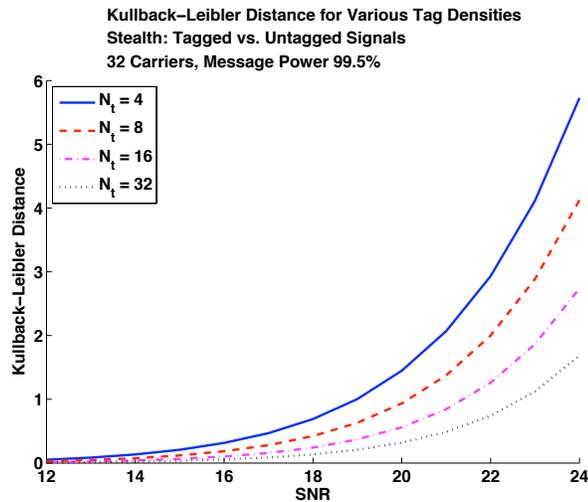


Figure 4.4: Stealth: KL distance between tagged and untagged signals for various  $N_t$  in AWGN. Increased tag spread (higher  $N_t$ ) decreases the KL distance and hence leads to better stealth.

### Impact

We have already established the effect of message power on the message BER in the previous chapter. Now we fix the message power and consider the effect of spreading the authentication.

Recall from Figure 4.3 that the more the authentication is spread, the less powerful each tag becomes. From the previous chapter, we know that the less powerful the tag, the better the BER. On the other hand, if we concentrate the tag power in only a few symbols, there are more symbols that do not experience interference from the tag. It may not be immediately obvious whether it is better to spread out the tags, and thereby slightly increase the BER of most symbols, or to concentrate the tags, and thereby increase the BER of a few symbols significantly while leaving the others unperturbed.

Figure 4.5 shows that for low message powers, the tag spread plays a signifi-

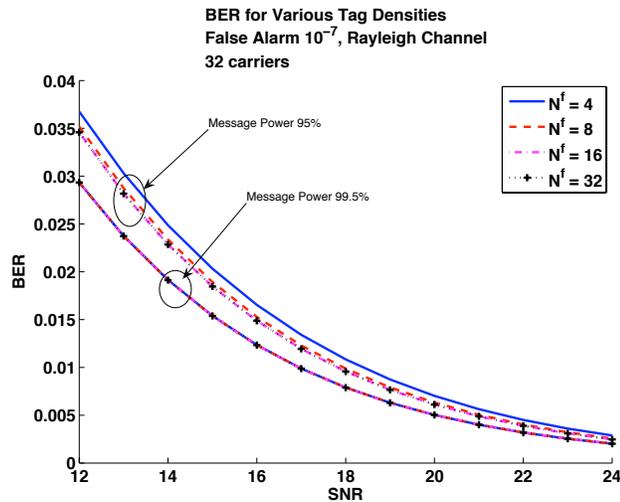


Figure 4.5: Stealth: BER for various  $N_t$  in AWGN. Increased tag spread (higher  $N_t$ ) decreases the BER for a fixed message power, though the effect diminishes as the message power increases.

cant role on the BER, where increased spreads benefit the BER. However, with higher message powers the spread no longer matters, as the BER curves nearly coincide. More importantly, the BER curves are so close that the packet outage probabilities are only slightly perturbed by the presence of the authentication.

### 4.3.2 Robustness

We now turn our attention to ability of the receiver to make correct authentication decisions.

Since we only authenticate unmodified messages, in the following we assume that the message is received correctly. Let us fix the message power and consider the effect of spreading the authentication tag over many symbols. In Section 4.2.4 we have seen that increasing  $N_t$  decreases the variance and kurtosis (tails) of the distribution which should improve authentication performance. Intuitively,

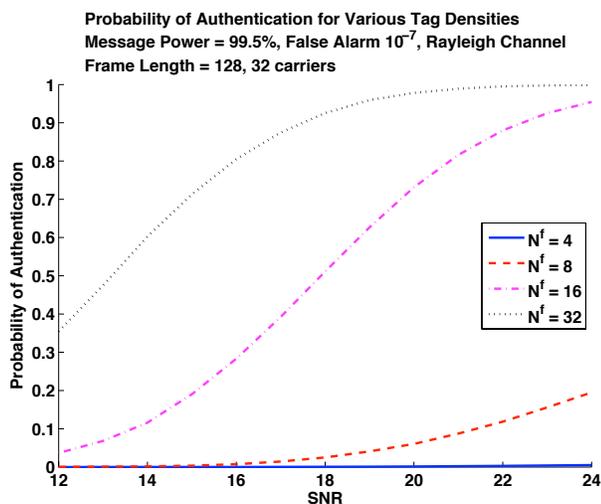


Figure 4.6: Robustness: Authentication probabilities for various tag spreads  $N_t$  in the Rayleigh fading channel. Spreading the tag over more symbols yields higher authentication probability and hence better robustness.

spreading the tag over many independent carriers decreases the probability of an outage (unacceptably low SNR) and hence is favorable for performance.

Figure 4.6 demonstrates that  $N_t$  does indeed play a major role in the authentication performance. It is clearly better to transmit many lower-powered tag symbols rather than a few higher-powered tag symbols.

The availability of multiple carriers also introduces the unwelcome effect of inter-carrier interference (ICI). As seen in Section 4.2.2, ICI is introduced when the oscillators of the transmitter and receiver are mismatched in frequency.

Figure 4.7 shows the authentication performance in the midst of frequency offset. Note that the effect of the frequency offset cannot be modeled simply as a SNR reduction when the offset  $\varepsilon$  is large, but for small  $\varepsilon$  it is a reasonable approximation. For small  $\varepsilon$ , the effect on the authentication performance is small and may be overcome through the methods introduced in this and the previous chapter: increasing tag power or length, placing the authentication over multiple

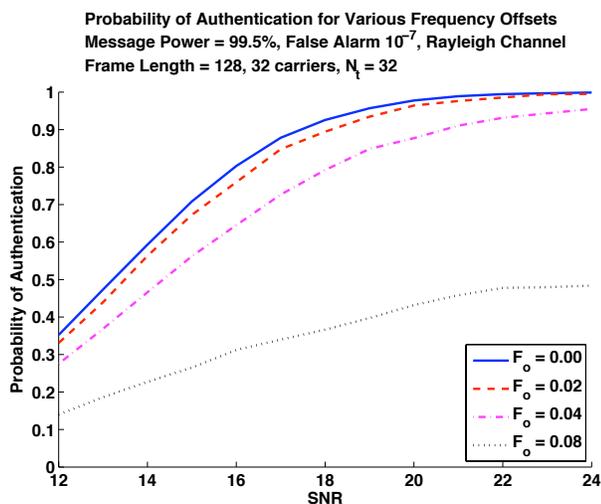


Figure 4.7: Robustness: Authentication probabilities for various frequency offsets  $\varepsilon$  in the Rayleigh fading channel. Small offsets inflict a manageable decrease in authentication performance.

frames, or increasing the spread of the tag.

### 4.3.3 Security

The security of the authentication is measured by how easily Eve can recover the secret key. First, we will consider the equivocation of the authentication tags in the worst case scenario where Eve knows exactly where the tags are located. Then, we will remove that assumption so that Eve must determine where the tag symbols are non-zero before she attempts to extract any information. If she extracts information where none is present, she can poison her cache of previously-collected information.

## Equivocation

When the tag is observed through a noisy channel, it leads to positive key equivocation. Suppose that the authentication tag is composed of  $M$  bits. For example, with  $N_t = 32$  and  $N^f = 4$  there are 128 symbols. With 4QAM there are therefore  $M = 256$  bits.

The equivocation of the authentication tag depends on the bit error rate that it is observed with. Suppose that the authentication tag  $\mathbf{t}$  is composed of  $M$  bits and is observed with i.i.d. bit errors with probability  $p^t$ . We can calculate the tag equivocation  $H(\mathbf{t}|p^t)$  by iterating through the number of bit errors the tags can contain (between 0 and  $M$ ). The probability of observing  $n$  errors in a length  $M$  tag with bit error probability  $p^t$  is

$$Pr(p^t, n, M) = (p^t)^n(1 - p^t)^{M-n} \quad (4.101)$$

The tag bit error probability  $p^t$  depends on the system parameters as described in Section 4.2.3. For example, the system using a hierarchical 4/16-QAM constellation observed through a Rayleigh frequency-selective channel is parametrized with tag power  $(\rho^t)^2$  and average SNR  $\gamma$ . The resulting tag bit error probability is given in equation (4.69).

Since tags with the same number of i.i.d. bit errors have the same probability of occurring (and there are  $\binom{M}{n}$  length  $M$  tags with  $n$  errors), the tag equivocation is

$$H(\mathbf{t}|p^t) = \sum_{\mathbf{t} \in \mathcal{T}} Pr(\mathbf{t} = t|p^t) \log_2 \frac{1}{Pr(\mathbf{t} = t|p^t)} \quad (4.102)$$

$$= \sum_{n=0}^M \binom{M}{n} Pr(p^t, n, M) \log_2 \frac{1}{Pr(p^t, n, M)} \quad (4.103)$$

where  $Pr(\cdot, \cdot, \cdot)$  is defined above in equation (4.101). We note that there is an abuse of notation in this equation in that  $p^t$  is actually a constant; it does not

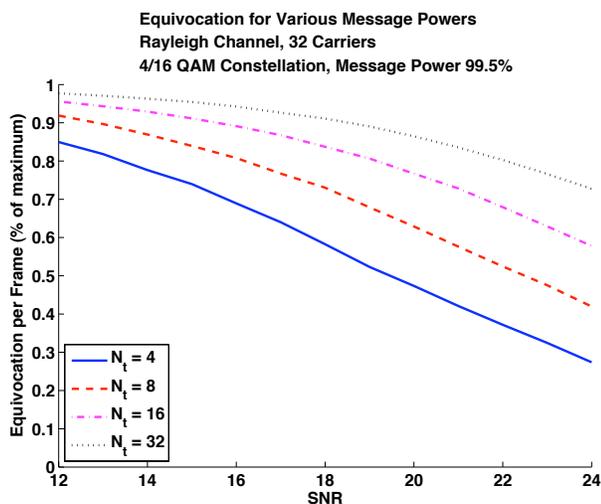


Figure 4.8: Security: Equivocation of the authentication tags for various tag spreads  $N_t$  in the Rayleigh fading channel. Spreading the tag over more symbols yields higher tag equivocation and hence better security.

depend on the particular tag.

Since the different tag spreads yield different bit-length tags, we normalize the equivocation as a percentage of the maximum equivocation for comparison. For example, when a 256-bit tag is observed with 128 bits of equivocation, it is equivalent to saying it is observed with 50% of the maximum equivocation. Figure 4.8 shows the tag equivocation over a Rayleigh block fading channel for various tag spreadings. It is clearly beneficial to spread the tag over as many carriers as possible.

Next we consider the effect of ICI on the equivocation. Based on previous results, we expect that increased ICI to favorably obscure the tag from the adversary. Figure 4.9 confirms the intuition.

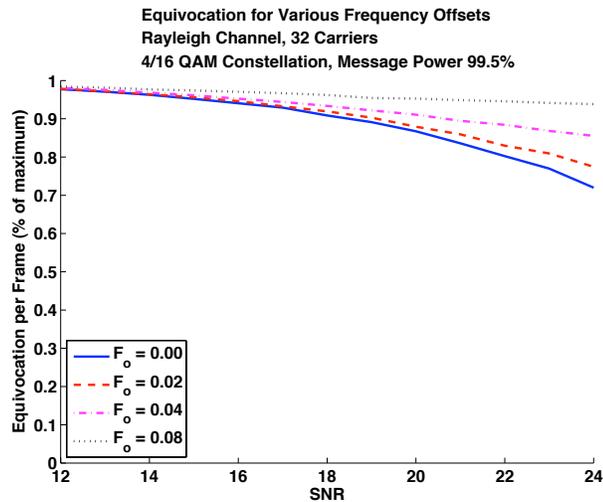


Figure 4.9: Security: Equivocation of the authentication tags for various frequency offsets  $\varepsilon$  in the Rayleigh fading channel. Increased  $\varepsilon$  (worse ICI) yields higher tag equivocation and hence better security.

### Poisoning

When we set the density of the tag symbols  $N^t/N$  below 1, some symbols do not contain any tag information. Therefore when gathering information about the secret key, the adversary needs to be careful not to extract information where there is none to be extracted. When false information is accumulated and accepted as true, the information is *poisoned*. This potentially leads Eve to the wrong conclusion about the tag. Therefore Eve should be careful to consider that her information is potentially false.

Eve must be able to do two things in order to recover tag, and hence key, information:

1. Decide the presence of a tag symbol
2. Decode the content of the tag symbol

The ability of the adversary to determine the location of the tag symbols is

limited by the stealth of the system. Suppose that when the symbol is untagged, the adversary falsely identifies it as tagged with false alarm<sup>2</sup> probability  $\alpha$ . When the symbol is tagged, the adversary correctly identifies it with probability  $1 - \beta$  which is dependent on the power and structure of the tag symbols.

There are  $\tilde{N} = NN^f$  total symbols, of which  $\tilde{N}^t = N^tN^f$  contain superimposed tag symbols.

Suppose that Eve collects tag information only from symbols that she knows for certain are tagged (that is, there are no false alarms and  $\alpha = 0$ ). By testing each of the  $\tilde{N}$  symbols, she can determine the presence of up to  $\tilde{N}^t$  tag symbols. The probability of detecting exactly  $x$  tag symbols is the binomial probability

$$Pr(x) = B(x; \tilde{N}^t, 1 - \beta) = \binom{\tilde{N}^t}{x} (1 - \beta)^x \beta^{\tilde{N}^t - x} \quad (4.104)$$

where the miss detection probability  $\beta$  is depends on the SNR  $\gamma$  and tag power allocation  $\rho^t$ . From the previous sections, we know that as either  $\gamma$  or  $\rho^t$  go down,  $\beta$  increases, making it more difficult for the presence of tag symbols to be detected.

Suppose that Eve is able to decide the presence of  $x$  tag symbols that contain  $x_b$  bits, e.g., for 4-QAM symbols we have  $x_b = 2x$ . Further assume that the tag bit error probability is  $p^t$ . Then the tag equivocation depends on how many of those  $x_b$  bits are received with error:

$$H(\mathbf{T}_x | p^t) = \sum_{n=0}^{x_b} \binom{x_b}{n} Pr(p^t, n, x_b) \log_2 \frac{1}{Pr(p^t, n, x_b)} \quad (4.105)$$

where  $Pr(p^t, n, M)$  is the probability of having  $n$  errors out of  $M$  bits when the bit error probability is  $p^t$  as defined in equation (4.101).

---

<sup>2</sup>Keeping the notation of detection theory,  $\alpha$  is the probability of false alarm (type I error) while  $\beta$  is the probability of a missed detection (type II error). See Chapter 1 for further elaboration.

When Eve observes the tags with bit error probability  $p^t$ , the tag equivocation thus depends on  $x$ , the number of tag symbols identified

$$H(\mathbf{T}_X|p^t) = \sum_{x=0}^{\tilde{N}^t} Pr(X = x)H(\mathbf{T}_{X=x}|p^t) \quad (4.106)$$

where  $Pr(\cdot)$  is the probability of detecting the presence of  $x$  tag symbols (equation (4.104)) and  $H(\cdot|\cdot)$  is the equivocation when  $x$  symbol are detected (equation (4.105)).

Figure 4.10 shows the effect of unknown tag positions on the key information gained by the adversary. We note that baseline performance is given when  $\beta = 0$ , which is the situation where the adversary knows the position of the entire tag precisely. However, in actuality the stealth requirements of the authentication (Section 4.3.1) force the detection probability to be low when the adversary sets her false alarm probability  $\alpha = 0$ . We see from the figure that this uncertainty is able to drastically reduce the amount of tag information available to Eve. Clearly, her performance is bounded by reliability of her tag detection, and we therefore clearly see the relationship between the stealth and security of the authentication.

## 4.4 Power Allocation

Suppose that Alice and Bob have channel state information (CSI). Then the transmitter can optimize the power loading across carriers to improve the message rate. It is well known that the water-filling power allocation maximizes the message rate for parallel Gaussian channels [38]. Thus when no authentication tag is transmitted, the optimal power allocation is given by

$$P_k = (\nu - N_k)^+ \quad (4.107)$$

$$1 = P = \sum_k (\nu - N_k)^+ \quad (4.108)$$

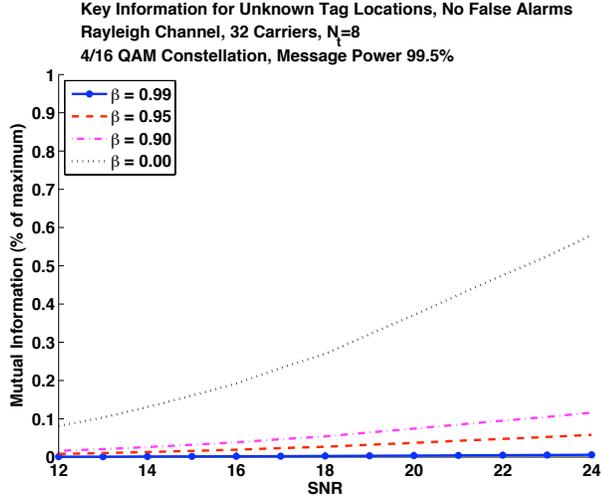


Figure 4.10: Security: Key information gained when tag positions are unknown in the Rayleigh fading channel. False alarm probability is zero. The higher the miss detection probability the better the security.

where  $P_k = \rho(k, k)^2$ ,  $P = \|\rho\|^2$  and  $N_k = \sigma_w^2 / |H(k, k)|^2$ . We assume that  $\rho$  is given and that the allocations  $\rho^s, \rho^t$  satisfy equation (4.13), i.e., the total power per carrier for tagged and untagged signals is equal. We require this for stealth purposes: if the power spectrum of the signal is different it is easy for the adversary to detect the anomaly.

For brevity in the sequel, we denote the per-carrier powers by  $P_k^s = \rho^s(k, k)^2$ ,  $P_k^t = \rho^t(k, k)^2$  and the total power constraints by  $P^s = \|\rho^s\|^2$ ,  $P^t = \|\rho^t\|^2$ .

In the authentication system, we transmit message and tags simultaneously, so the question becomes how to best allocate the power between message and tag on a per-carrier basis given  $P^s$  and  $P^t$  (the percentage of power used for the message and tag).

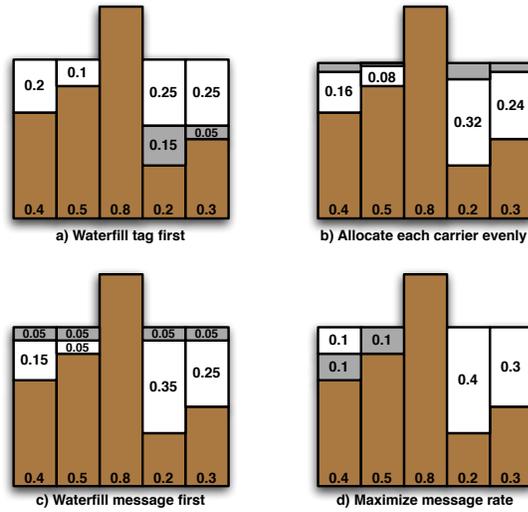


Figure 4.11: Power allocation strategies. Base bars represent noise power on the carriers, white bars represent message power, and lightly shaded bars represent tag power. Power allocation is 80% message and 20% tag ( $P^s = 0.8, P^t = 0.2$ ).

#### 4.4.1 Strategies

The water-filling allocation given above maximizes the message rate of the system when not tag is transmitted. We consider four power allocation strategies that are easy to implement and have different merits that we will compare in the next section. Figure 4.11 illustrates the allocations.

By design each of the power allocation strategies yields the same signal power per carrier as the untagged signal. This is done for stealth purposes: an abnormal power spectrum can be easily detected and flagged as anomalous by adversaries.

### Waterfill Tag, then Message

First, allocate the tag powers  $P_k^t$  by water-filling with the power budget  $P^t$ .

$$P_k^t = (\nu_t - N_k)^+ \quad (4.109)$$

$$P^t = \sum_k (\nu_t - N_k)^+ \quad (4.110)$$

Then, treating the tag power as noise, allocate the message powers  $P_k^s$  by water-filling with the power budget  $P^s$ .

$$P_k^s = (\nu_s - N_k - P_k^t)^+ \quad (4.111)$$

$$P^s = \sum_k (\nu_s - N_k - P_k^t)^+ \quad (4.112)$$

This strategy is shown in Figure 4.11a. In this case, the message always occupies at least as many carriers as the tag.

### Evenly allocate

First we determine the signal powers  $P_k$  that will be used on each carrier using the total power budget  $P$  by using equations (4.107) and (4.108). Then, using the message and tag power allocation we calculate the message and tag powers per carrier

$$P_k^s = P^s P_k \quad (4.113)$$

$$P_k^t = P^t P_k \quad (4.114)$$

This strategy is shown in Figure 4.11b. The proportion of message to tag power is consistent for each carrier with non-zero signal power. In this case, the message always occupies the same carriers as the tag.

### Waterfill Message, then Tag

First, allocate the message powers  $P_k^s$  with the power budget  $P^s$ .

$$P_k^s = (\nu_s - N_k)^+ \quad (4.115)$$

$$P^s = \sum_k (\nu_s - N_k)^+ \quad (4.116)$$

Then, treating the message power as noise, allocate the tag powers  $P_k^t$  with the power budget  $P^t$ .

$$P_k^t = (\nu_t - N_k - P_k^s)^+ \quad (4.117)$$

$$P^t = \sum_k (\nu_t - N_k - P_k^s)^+ \quad (4.118)$$

This strategy is shown in Figure 4.11c. In this case, the tag always occupies at least as many carriers as the message.

### Maximization of Message Rate

Consider the message capacity of the  $k^{th}$  carrier. With the message and tag allocations  $P_k^s$  and  $P_k^t$ , it is

$$C_k^s = \frac{1}{2} \log \left( 1 + \frac{P_k^s}{N_k + P_k^t} \right) \quad (4.119)$$

Note that the tag acts as additional noise to the message. With the water-filling allocation (4.107), we may simplify this equation to

$$C_k^s = \begin{cases} \frac{1}{2} \log \left( \frac{\nu}{N_k + P_k^t} \right) & P_k^s > 0 \\ 0 & \text{otherwise} \end{cases} \quad (4.120)$$

Suppose we wish to allocate power across carriers such that the message rate is maximized. From (4.120) is clear that carriers with zero message power have no contribution to the capacity. Thus we remove the carriers with  $P_k^s = 0$  from consideration, and for brevity write  $\sum_k$  to mean  $\sum_{k|P_k^s > 0}$ .

The constrained optimization problem is

$$\max_{\mathbf{P}^t} \sum_k C_k^s \quad (4.121)$$

with the constraints

$$\sum_k P_k^t = P^t = 1 - \|\rho^s\|^2 \quad (4.122)$$

$$P_k^t \geq 0, \forall k \quad (4.123)$$

$$P_k^t \leq P_k, \forall k \quad (4.124)$$

We use the Lagrange method to solve the problem. The objective function is

$$\begin{aligned} J(\mathbf{P}^t) &= \sum_k C_k^s \\ &+ \lambda(P^t - \sum_k P_k^t) \\ &+ \sum_k \mu_k^- P_k^t \\ &+ \sum_k \mu_k^+ (\nu - (N_k + P_k^t)) \end{aligned} \quad (4.125)$$

Since the cost function is concave and each constraint is linear, the KKT conditions are necessary and sufficient to solve the problem. The KKT conditions are

$$\frac{\partial J(\mathbf{P}^t)}{\partial P_k^t} = 0, \forall k \quad (4.126)$$

$$\lambda(P^t - \sum_k P_k^t) = 0, \lambda \geq 0 \quad (4.127)$$

$$\mu_k^- P_k^t = 0, \mu_k^- \geq 0, \forall k \quad (4.128)$$

$$\mu_k^+ (\nu - (N_k + P_k^t)) = 0, \mu_k^+ \geq 0, \forall k \quad (4.129)$$

Setting the derivative to zero (4.126), we have

$$-\frac{1}{2} \frac{1}{N_k + P_k^t} + \mu_k^- - \mu_k^+ = \lambda \quad (4.130)$$

Case 1:  $P_k^t = 0$ . Then  $\mu_k^- > 0, \mu_k^+ = 0$

$$-\frac{1}{2} \frac{1}{N_k} + \mu_k^- = \lambda \quad (4.131)$$

Case 2:  $0 < P_k^t < \nu - N_k$ . Then  $\mu_k^- = 0, \mu_k^+ = 0$

$$-\frac{1}{2} \frac{1}{N_k + P_k^t} = \lambda \quad (4.132)$$

Case 3:  $P_k^t = \nu - N_k$ . Then  $\mu_k^- = 0, \mu_k^+ > 0$

$$-\frac{1}{2} \frac{1}{\nu} - \mu_k^+ = \lambda \quad (4.133)$$

Ambiguities remain since there are multiple power allocations that will satisfy the above equations. To proceed further we use the following lemma. This lemma indicates that to maximize the total capacity of two independent channels, it is better to add any interference (e.g., tags) in the noisier of two channels.

**Lemma 1** For  $\Delta > 0$ ,

$$\begin{aligned} & \frac{1}{2} \left[ \log \left( \frac{\nu}{N_1 + \Delta} \right) + \log \left( \frac{\nu}{N_2} \right) \right] \\ & > \frac{1}{2} \left[ \log \left( \frac{\nu}{N_1} \right) + \log \left( \frac{\nu}{N_2 + \Delta} \right) \right] \end{aligned}$$

if and only if  $N_1 > N_2$ .

Together with the KKT conditions, it is clear that the optimal strategy places the tag power in the highest noise carriers. The sum power of the tags are distributed in cases 2 and 3. With the lemma, a single carrier can satisfy case 2. The other carriers are either dedicated to message or tag. It is easy to check that the following algorithm yields an optimal solution (it may not be unique):

1. Define (descending) order statistics  $t_1, \dots, t_K$  such that  $N_{(t_1)} \geq N_{(t_2)} \geq \dots \geq N_{(t_K)}$

2. Initialize  $k = \arg[\min_l(\nu - N_{(t_l)}) > 0]$ .
3. While  $k \leq K$ 
  - $P_{(t_k)}^t = \min \left( \left( T - \sum_{l < k} P_{(t_l)}^t \right)^+, \nu - N_{(t_k)} \right)$
  - $k = k+1$

This strategy is shown in Figure 4.11d. The algorithm greedily places the tag power in the carriers with the highest noise until there is not enough power to entirely occupy any of the remaining carriers. At that point, the remaining tag power is placed in the next noisiest carrier. Note that in this strategy, at most one carrier is used to signal both message and tag.

#### 4.4.2 Capacity

The message and tag capacities are respectively

$$C^s = \sum_k \frac{1}{2} \log_2 \left( 1 + \frac{P_k^s}{N_k + P_k^t} \right) \text{ bits} \quad (4.134)$$

$$C^t = \sum_k \frac{1}{2} \log_2 \left( 1 + \frac{P_k^t}{N_k} \right) \text{ bits} \quad (4.135)$$

Note that when recovering the message, the receiver treats the tag as additive noise. However, when the tag is recovered, the message is known since it is transmitted with higher power. This is the superimposed method outlined by Cover [39].

The message capacities for the four strategies are shown in Figure 4.12. In order, the best strategies are 4,3,2, and then 1. The message rate that arises from greedily using the highest SNR carriers for message power is vastly superior to those from other strategies. Of the four, the worst strategy in terms of message capacity is to place the tag in the highest SNR carriers.

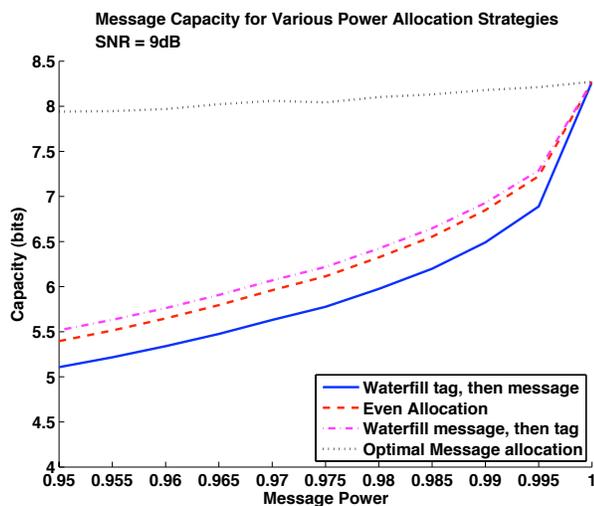


Figure 4.12: Message Capacity

The tag capacities for the four strategies are shown in Figure 4.13 when the SNR = 9dB. In order, the best strategies are 1,2,3 and then 4. The performance of the strategies with respect to tag capacities is reversed compared to the message capacities. While the optimal message allocation (strategy 4) is the best in terms of message capacity, it is severely penalized when it comes to tag capacity. For authentication, maximizing the tag rate is not necessarily a priority because tags are low rate compared to messages.

### 4.4.3 Authentication Metrics

In the following simulations, we assume the same basic parameters as before. The Monte Carlo simulation parameters are summarized in Table 4.2.

A range of false alarm probabilities was simulated, and the pictured false alarm probability was chosen to give an example of a reasonable operating point. We note that as the false alarm probability becomes smaller, the power of the authentication test does not change much because the tails of the distribution

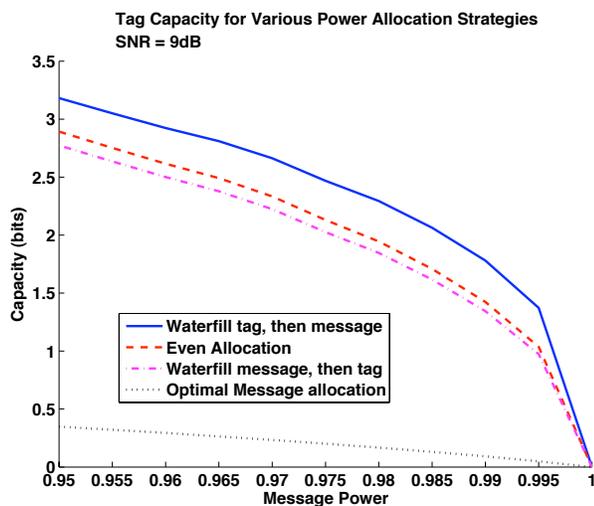


Figure 4.13: Tag Capacity

under  $H_0$  (4.73) are very small. With suitably low false alarms, the authentication decisions may be trusted with high probability.

## Stealth

The stealth of the authentication system can be measured by its message throughput and by its BER. We consider each in turn.

### Throughput

The message throughput for various policies is shown in Figure 4.14. The throughput using strategy 4 (the optimal message allocation) is consistently high when the message power is high ( $P^s$  close to  $P = 1$ ). The other strategies are more noticeably affected by the decrease in message power. However, the throughputs are not affected in the same way.

Strategies 2 and 3 offer reasonably high throughputs when the message power is high. There is little difference between the two, though Strategy 2 is marginally better.

Finally, strategy 1 has the lowest throughput of the four power allocation

Table 4.2: Simulation parameters for the multi-carrier, perfect CSI case

Channel Model	Rayleigh block fading
Noise Model	AWGN
# Carriers	32 (4 taps)
Channel State Information?	Yes
Modulation	BPSK: SNR $\leq$ 7dB 4-QAM: SNR $>$ 7 dB 16-QAM: SNR $>$ 12 dB 64-QAM: SNR $>$ 17 dB
Channel Estimate Method	Known
# Pilot Symbols	1 OFDM symbol per frame
Frame Length	4 OFDM symbols
False Alarm Probability	$10^{-7}$
# Monte Carlo Samples	$2^{14}$

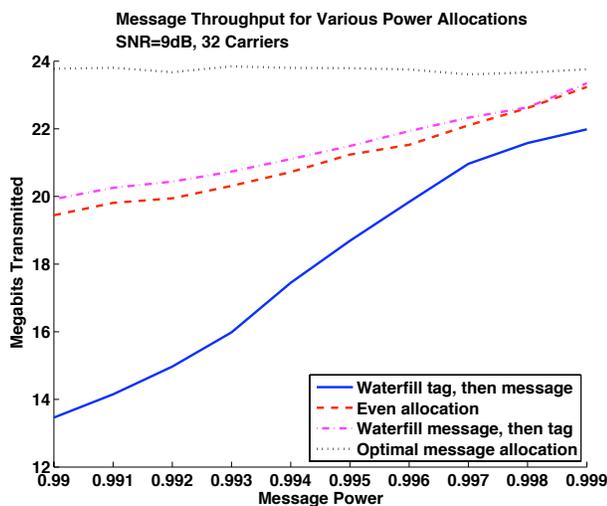


Figure 4.14: Throughput for various strategies. Frame length = 32 symbols. Average SNR = 9 dB.

strategies. By signaling the tag over the highest SNR carriers, the effective message is lowered, thus having a substantial impact on throughput when  $P^s$  is not very close to  $P = 1$ .

### Message BER

When the authentication tag is present, power is necessarily allocated away from the message, and hence the message BER increases. The impact of the authentication tag varies depending on the power allocation strategy.

Figure 4.15 shows the increase in BER for various strategies. The BER is the least affected when the message power is near 1. Of the strategies, the optimal message allocation has the best stealth: the BER is the least impacted for all message powers.

### Robustness

The robustness of the authentication system is given by its probability of authentication for a given false alarm probability. We compare the effect of frame

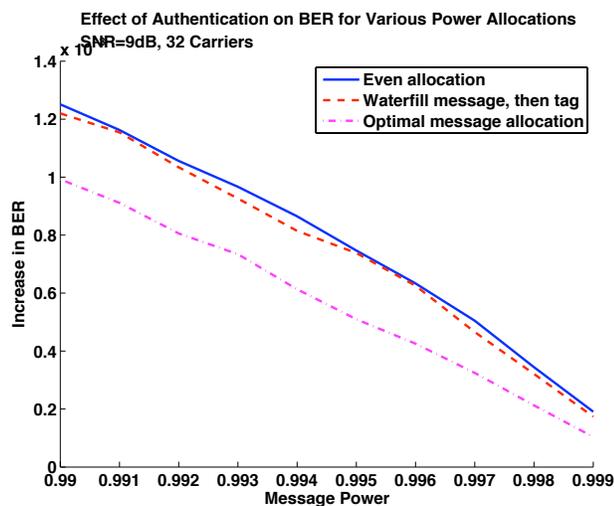


Figure 4.15: Stealth for various strategies. Frame length = 32 symbols. Average SNR = 9 dB.

lengths as well as the effect of various power allocation strategies.

Figure 4.16 shows that the choice of policy can greatly impact the robustness of the authentication system. The best performing strategy is to allocate water-fill the tag first before water-filling the message. Strategies 1-3 have approximately equal performance, but strategy 4 performs much worse.

Since strategy 4 places the tag at the lowest SNR carriers, the tag detection does not receive much benefit from any frequency diversity. The tags are placed in the highest noise regions by design in order to maximize the message throughput, and as a result the authentication performance suffers.

## Security

The stealth of the authentication system is given by the tag equivocation of the unaware or adversarial receiver. We compare the equivocation for the policies as shown in Figure 4.17. Note that the maximum equivocation of the tag is determined by the number of symbols in the frame. With a 32-carrier system

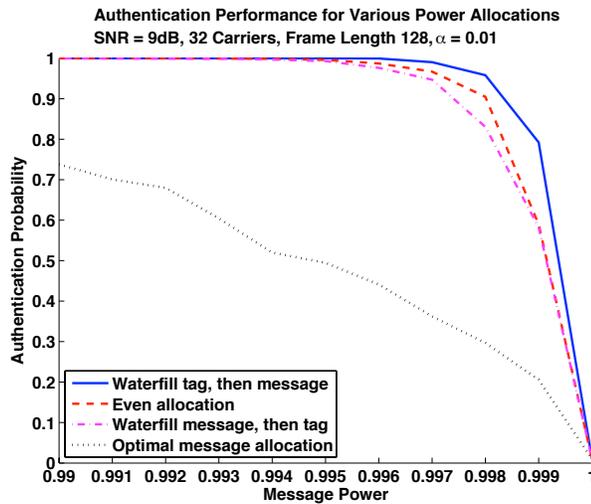


Figure 4.16: Robustness for various strategies. Frame length = 32 symbols. Average SNR = 9 dB. False alarm probability  $\alpha = 0.01$ .

and 4 OFDM symbol packets, the maximum equivocation is 128 bits.

Clearly the power allocation that maximizes message capacity also maximizes the tag equivocation among the policies. However, from the previous section we see that this allocation also performs the worst in terms of authentication robustness. The remaining two policies result in very similar equivocation, demonstrating that proportionally allocating power between message and authentication is a reasonable strategy with little tradeoff. As before, higher SNR situations reduce the tag equivocation.

## 4.5 Conclusion

We have extended the authentication framework for multiple-carrier systems and have identified that the placement of the tags across the carriers is an important performance parameter. Spreading the tags across as many tags as possible is beneficial to stealth and robustness, but may not be possible due to the limitations

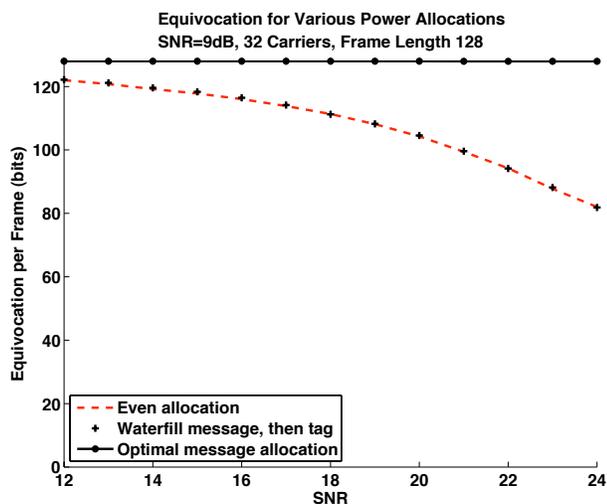


Figure 4.17: Tag equivocation for various strategies. Frame length = 32 symbols. Average SNR = 9 dB. False alarm probability  $\alpha = 0.01$ .

of hardware. However, we have also found that leaving some symbols untagged makes it much more difficult for the adversary to obtain key information.

The multiple-carrier environment also leads to undesirable effects such as frequency offset and increased channel estimation errors. However, we have shown that for small errors they are tolerable and may be overcome through use of increased tag energies or increased tag spread.

When channel state information is known to the transmitter, we demonstrated that the allocation of the tag power plays a very important role in terms of maintaining stealth and robustness. While it is possible to place tag energy so maximize the message throughput, it is unusable for authentication. Allocating power between message and tag at a constant ratio per carrier is shown to be a reasonable compromise between the metrics and is also the lightest in terms of computation.

## Chapter 5

### Experimental Results

#### 5.1 Overview of Contributions

In this section we describe the experimental testbed that was created to validate the theory put forth in chapters 2 and 3. We detail the hardware and software components of the testbed, describe the test scenarios and collected data, and finally discuss the results.

- The physical layer authentication is implemented over a software radio testbed and demonstrated to perform well
- The stealth of the authentication is reasonable and benefits from the time-variation of the channel
- The robustness of the authentication is shown to have a strong relationship with the tag energy. The reliability of the authentication is shown to be high for reasonable power allocations.
- The authentication is shown to be resistant to false alarms when incorrect keys are used by the transmitter or receiver. This bolsters the security analysis of the authentication scheme.

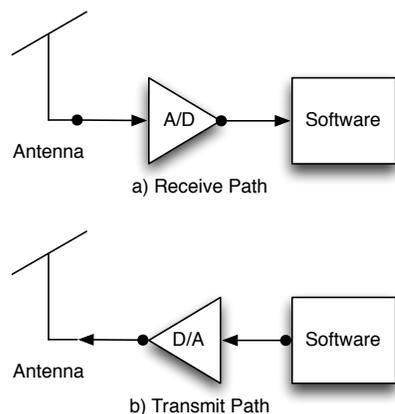


Figure 5.1: A fundamental concept of software defined radios is the placement of software as close as possible to the antennae. Only an analog-to-digital converter (ADC) separates the software from the antenna in the receive path (a), while a DAC is present in the transmit path (b).

## 5.2 Experiment Setup

The authentication scheme was implemented on GNU Radio - a software defined radio (SDR) platform that is in active development by the open source community. The primary concept of software radio is to have the software as close as possible to the antenna as shown in Figure 5.1. Compared with traditional radios where modulations and codes are defined with special circuitry, SDR shifts the computational load from hardware to software. With the increase in processing power and the associated decrease in cost, SDR is becoming a more and more viable solution for powerful and adaptable radios. Practically speaking, the SDR paradigm increases the speed and ease of prototyping, testing, and configuring new radios.

For our experiment, we do not modify the hardware, but we make extensive software-side extensions to implement the authentication. However, since

the hardware imposes limitations on the authentication (Chapters 2 and 3), we first detail the relevant hardware specifications. Then, we detail our software implementation of the physical layer authentication scheme over the GNU radio platform.

### 5.2.1 Hardware Capabilities

The software interfaces with the radio transceiver via USB (universal serial bus) interface. The radio transceiver in our experiment is the Universal Software Radio Peripheral (USRP, pronounced "*usurp*") which is the most popular and commonly available peripheral used by the GNU Radio project. As seen in Figure 5.2, the USRP consists of a USB interface, an FPGA (field-programmable gate array), ADCs and DACs (analog-digital and digital-analog converters, respectively), and daughterboards. The daughterboards are responsible for the frequency tuning and conversion between IF and RF (intermediate and radio frequencies, respectively), and are swappable for flexible configuration. In the following we detail the signal receive path to highlight the design of the hardware.

#### Daughterboard RFX2400

The signal is captured by an antenna attached to an RFX2400 daughterboard. The RFX2400 is a 2.3-2.9GHz band transceiver with a 20MHz transmit/receive bandwidth. The received signal passes through a mixer to downconvert the signal to the IF<sup>1</sup>. Then, the signal is then amplified up to 70dB via AGC (automatic

---

<sup>1</sup>By converting signals to an IF rather than going directly between RF and baseband, the quality of the circuit can be vastly improved (by allowing use of crystal filters, for example). Receivers which do this are called superheterodyne for their use of the heterodyne principle which is based on the identity  $2 \sin(\theta) \sin(\phi) = \cos(\theta - \phi) - \cos(\theta + \phi)$ .

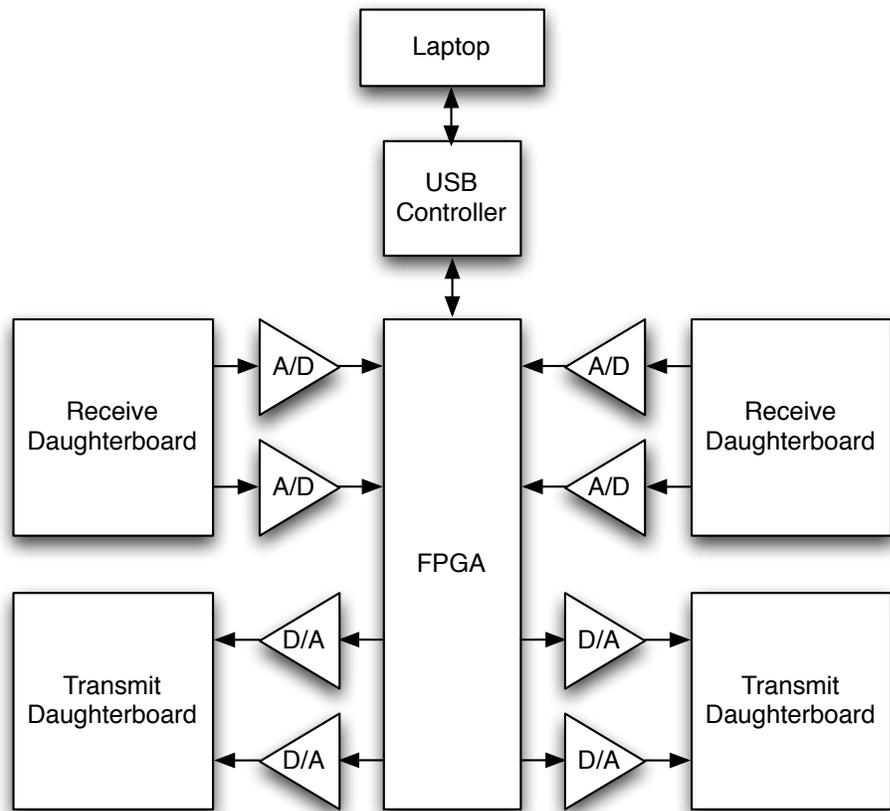


Figure 5.2: An overview of the hardware setup: the laptop is connected via USB to the USRP. The USRP consists of an FPGA responsible for up/down conversions, ADCs and DACs, and various plug-in daughterboards.

gain control) before being sent to the USRP motherboard.

## **USRP**

The USRP board has four 12-bit ADCs that are capable of processing up to 64 mega-samples per second from the daughterboards. Depending on configuration, these channels may contain either real or paired I/Q (in-phase and quadrature, respectively) samples.

The digitized samples are then sent to the FPGA<sup>2</sup>. The FPGA uses a numerically controlled oscillator (NCO) to convert the samples from IF to baseband. Then CIC (cascaded integrator-comb) filters are used to decimate the oversampled signal to lower the data rate. This paring down is the DDC (digital down conversion) and is necessary for transmission over the USB 2.0 interface. The resultant total bandwidth over all channels is limited to 32 megabytes/sec: 16-bit signed integers in I/Q format, i.e., 4-bytes per complex sample at 8 megasamples/sec. Of course, lower bandwidths are possible by setting the decimation factor, e.g.  $64 \text{ MHz}/250 = 256\text{kHz}$ . Finally, the samples are transmitted to the computer via USB.

The transmit path is essentially the reverse of the receive path. Digital samples arrive at the USRP via USB and are interpolated and up-converted to the IF. Then they are passed through DACs and sent to the daughterboard, where they are mixed to the RF, amplified, and transmitted over the antenna.

## **Laptops**

We use two identical 2.0 GHz Pentium M laptops with 512 MB RAM. Each runs Ubuntu Linux 7.04 (Feisty Fawn) with the GNU Radio software installed. The

---

<sup>2</sup>Altera Cyclone EP1C12 chip

software is extended for physical layer authentication capability as described in Section 5.2.2. One laptop controls the transmitter; the other controls the receiver.

### 5.2.2 Software Design

We modified the GNU Radio platform to add authentication at the physical layer. It was written with a combination of C++ and Python for a good tradeoff between processing speed and rapid prototyping. The signal processing blocks (e.g., filters, phase locked loops) were written in C++ and joined together in Python.

For this experiment we modified existing signaling blocks and also created our own. In the following, we detail the changes made to the transmit and receive paths.

#### Transmitter

Figure 5.3 shows a diagram of the transmitter. The original system takes the payload and constructs a packet around it. It adds a preamble, access code, header, and cyclic redundancy check (CRC) (Figure 5.4). The packet is then DBPSK-modulated, pulse shaped, and transmitted.

To implement the authentication, we made the following changes. We added a tag creation block that generates the authentication tag from the payload and a secret key. Then, the authentication tag was padded to align the tag with the message payload (Figure 5.4). The message packet and padded tag are scaled and superimposed - the padding ensures that only the message payload is perturbed and that the important header information is untouched. In general we may choose to perturb the entire packet. However, since the header may be used for synchronization or other important purposes, we chose not to alter it.

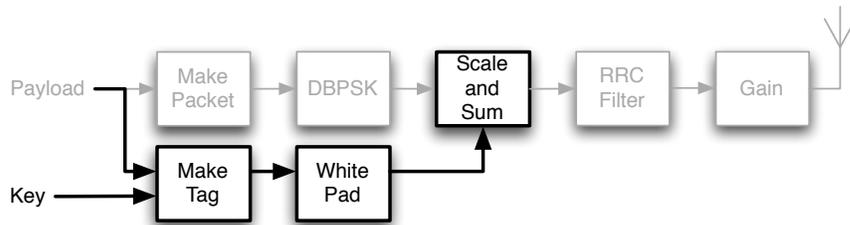


Figure 5.3: Transmitter signal path. Unmodified processing blocks are grayed out; modifications are darkened.

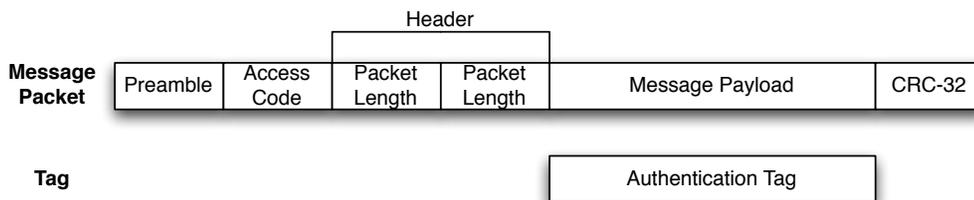


Figure 5.4: Packet format. Note that the tag has non-null information coincident with the packet payload; no other portion fo the packet is modified by the superposition.

In our implementation we use binary signaling for the authentication tag: we either increase or decrease the voltage of a payload symbol depending on each particular tag bit. This has the nice property of being easy to decode over DBPSK-modulated messages since the receiver only has to observe the symbol amplitude and not the symbol phase.

## Receiver

Figure 5.5 shows a diagram of the receiver. The receiver performs automatic gain control (AGC), root-raised cosine (RRC) filtering, timing (Mueller and Muller algorithm) and phase (Costas loop) synchronization before DBPSK demodulation.

In the original system, after digitization the receiver scans for the access code

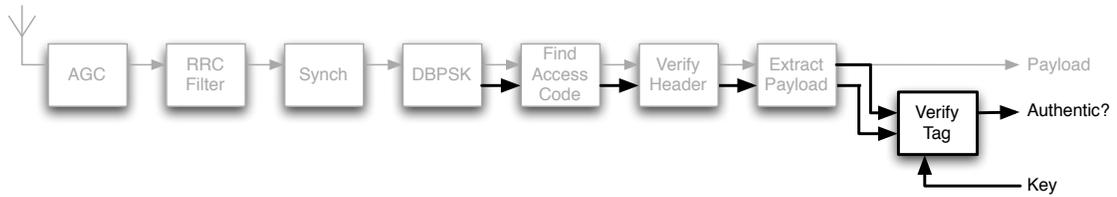


Figure 5.5: Receiver signal path. Unmodified processing blocks are grayed out; modifications are darkened.

that indicates the beginning of a packet. After finding the access code, it then verifies the integrity of the header and extracts the payload with CRC. The payload is then checked with the CRC. If the redundancy check fails the packet is discarded; otherwise it is accepted.

To obtain the tag, we modified the DBPSK module to return not only the demodulated symbols but the raw analog samples. The sampled signals are paired and together pass through each subsequent block until the payload is verified. If the payload passes the CRC check, the signals proceed to the tag detection block; otherwise the samples are discarded and no further processing is done.

With a successful CRC check, the sampled signal arrives at the tag detection block. The verified payload (symbols) and the receiver’s secret key are used to generate the authentication tag. The receiver takes the analog signal corresponding to the payload (Figure 5.4) and correlates them with the tag. When the correlation exceeds the threshold chosen to limit false alarms, the packet is deemed authentic and accepted.

### 5.3 Testing Procedure and Results

The transmit and receive stations were placed approximately 20 feet apart without a line of sight. The transceivers operate at 2.44 GHz to avoid strong inter-

ference from the campus wireless network and from cordless telephones.

The receiver continuously scans the channel for packets. The transmitter sends 48k ( $4 * 2^{10}$ ) packets at 500 kbps. We used two payload lengths: 128 and 192 bytes, of which 4 bytes are set aside as pilot symbols. For each packet length, we consider the following test scenarios (TS):

TS 1) The transmitter does not transmit any authentication.

TS 2) The transmitter superimposes the authentication on the packets but its secret key does not match that of the receiver.

TS 3) The transmitter superimposes the authentication on the packets and its secret key matches that of the receiver.

The receiver should reject the packets in cases 1 and 2, and only accept the packets in case 3. Accepting a packet in case 1 is the most innocuous false alarm. In case 2, accepting a packet leads to a security breach since the keys do not match. For cases 2 and 3, where the authentication is present, the experiments were repeated at 0.1, 0.2, 0.3, . . . , 1.0% authentication powers.

The following data sets (DS) were collected:

DS 1) Digitized signal samples

DS 2) Number of received packets (error-free)

DS 3) Number of authenticated packets

The interpretation of the data depends on the test scenario and are discussed in the following sections. Figure 5.6 gives a preview of how the data was processed.

		<b>Data Set</b>		
		1) Sampled Signal	2) # Received Packets	3) # Authenticated Packets
<b>Test Scenario</b>	1) No Authentication	<b>Stealth: <i>Presence</i></b>	<b>Stealth: <i>Impact</i></b>	<b>Security</b>
	2) Wrong Key			
	3) Correct Key		<b>Robustness</b>	

Figure 5.6: The test scenarios and data sets are used to evaluate the authentication system. The data collected in each test scenario is used to compute the stealth, robustness, or stealth metrics.

### 5.3.1 Stealth

We quantify the stealth of the authentication system based on the impact and presence of the authentication tags. These are measured by packet error rate and noise distribution, respectively. The packet error rate indicates the impact of the authentication on message recovery. The noise distribution indicates the detectability of the perturbation to the unaware receiver.

#### Impact of Authentication

The impact of the authentication upon the receiver is found by comparing the number of packets received without error (DS 2) between the scenarios when the authentication is absent versus when it is present (TS 1 vs. TS 2 and 3). Since there are many factors which affect how packets are dropped (e.g., not detected, failed header check, failed CRC) which may be due to time variation of the channel, we repeat the experiment multiple times. The impact of the

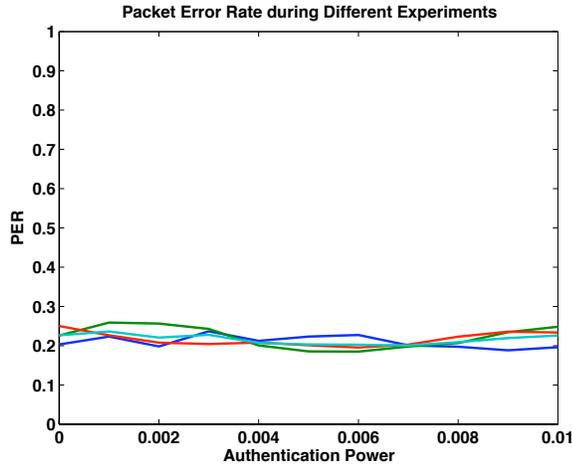


Figure 5.7: The packet error rate for various sample runs versus the power of the authentication signal. At low authentication powers, no significant deviation from the baseline packet error rate was observed. Each line represents a different test run.

authentication is found by determining how adding the authentication causes the packet error rate increase.

The observed packet error rates are shown in Figure 5.7. We observe no conclusive link between authentication power and packet error rate for the range of authentication powers tested. At such low authentication power, we suggest that the perturbation has a minimal impact and that the time-varying nature of the channel plays a much greater role on the packet error. This confirms the analysis in chapter 2.

Figure 5.8 is a snapshot of observed SNR across consecutive frames for various values of authentication power. The SNR values are obtained through the use of pilot symbols. We note that the SNR is not noticeably degraded with the addition of low power authentication tags; rather, the time variation of the channel plays a much larger role in the measured SNR. We calculate the 95%

confidence interval that the received signal does not contain authentication. In this particular snapshot, the majority of the packet SNR in three cases fall inside the 95% confidence interval. For those SNRs that fall outside of the interval, most are actually false alarms (when no authentication is transmitted).

### 5.3.2 Presence

The previous section established that the packet reception is minimally impacted when the authentication is injected at low power. Now we turn our attention to the distortion that is observed by the receiver. For each packet that is received correctly, we record the amplitude distortion (DS 1).

We study the noise by calculating the empirical cumulative distribution function (CDF) of the amplitude distortion over thousands of packets. The baseline distribution yields the noise characteristics of the channel under normal conditions, i.e., when no authentication is transmitted.

The presence of the authentication system is hidden when the resultant noise distribution at the receiver is close to the baseline. When the noise distribution is not close, its presence can be discovered through the use of goodness of fit tests. Assume that the receiver knows the baseline noise CDF, perhaps through training with a known transmitter. The receiver can compare the observed noise statistics with the baseline CDF in order to determine whether the signal is being perturbed. The tests operate at a user-specified probability of false alarm which is usually set very low to return useful detections.

Figure 5.9 shows the CDF for some representative authentication powers. The CDF are further apart when the authentication power increases. In theory, goodness of fit tests will therefore reject the distributions as being unequal when the authentication power is too high.

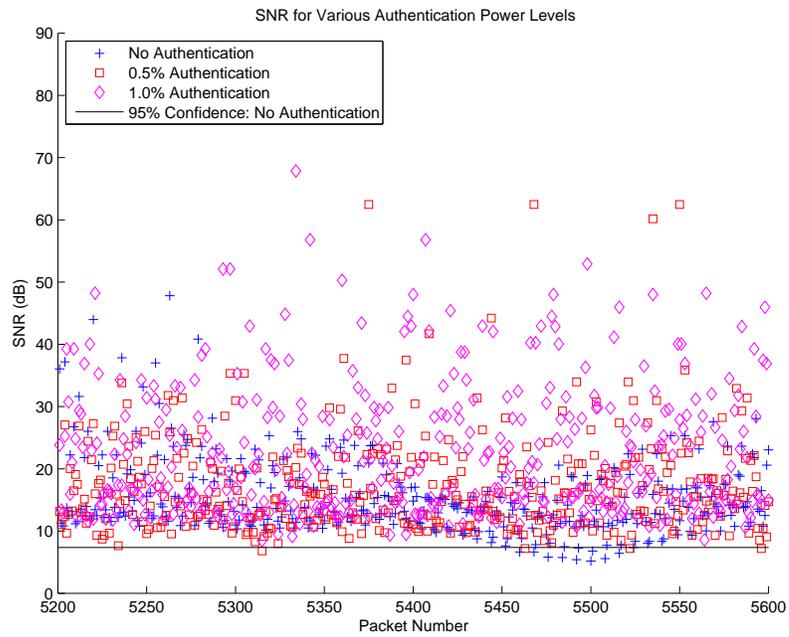


Figure 5.8: The observed SNR of tagged and untagged signals for a few consecutive packets. The majority of the packet SNR in three cases fall inside the 95% confidence interval for no authentication present in the signal; in this snapshot most of those that fall outside of it are actually false alarms.

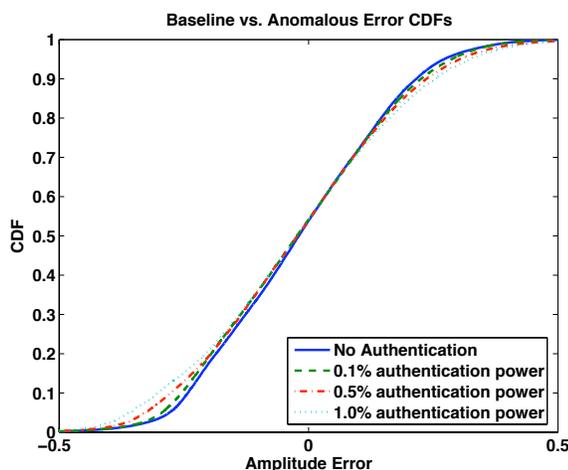


Figure 5.9: The observed CDF of the estimated noise for various authentication powers over thousands of packets. Larger authentication powers deviate more from the baseline CDF.

However, we found that the time-variation of the channels inhibits good performance of the tests. Using the Kolomogorov-Smirnov test with a 1% false alarm probability over a window of a few hundred packets, the receiver correctly flags the tagged signals as anomalous. However, the receiver also flags the untagged signals as anomalous - even though no authentication was being transmitted. Thus the receiver needs more powerful techniques to discriminate between pure noise and perturbation in noise - the KS test is not able to distinguish between tagged and untagged signals reliably.

The fact that it is difficult to discriminate using goodness of fit tests indicates that stealth may be further improved by time-multiplexing the authentication. That is, the receiver is faced with a more difficult anomaly detection problem when the authentication tags are injected into some, but not all, of the packets.

Table 5.1: Authentication Probability

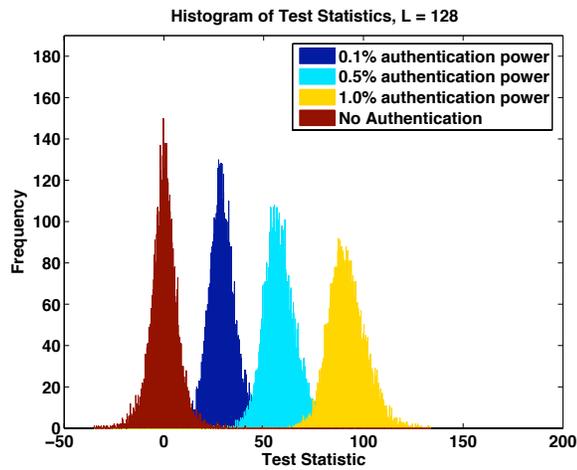
	Tag Power			
	0	0.001	0.005	0.010
L = 128	0.001	0.391	0.999	1.000
L = 192	0.001	0.973	1.000	1.000

### 5.3.3 Robustness

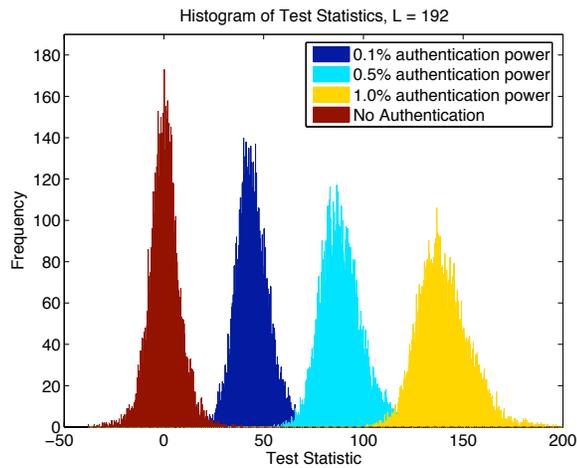
We quantify the robustness of the authentication system by its authentication probability for a fixed false alarm probability. We have the transmitter and receiver share the same key (TS 3) and analyze the number of authenticated packets (DS 3) for both 128 and 192-byte payloads.

The detection probabilities are found in Table 5.1. With the same power allocation, longer authentication tags have more energy and thus result in higher quality decisions. For example, increasing the payload from 128 to 192 bytes increases the detection probability from 39% to 97%.

The test statistics with 128-byte payloads are shown in Figure 5.10(a) for various authentication powers. The experiment is repeated for 192 byte payloads as shown in Figure 5.10(b). The statistics are clearly separated from the untagged signal case (no authentication transmitted), even for very low authentication power. Increasing the perturbation length increases the energy, and hence the performance of the authentication improves as well. We see that there is a clear relationship between the energy of the authentication and its performance.



(a)  $L = 128$  Histogram



(b)  $L = 192$  Histogram

Figure 5.10: Test statistic histograms for various length payloads. Longer payloads yield better signal separation and hence better authentication performance.

### 5.3.4 Security

We measure the security of the authentication system by observing the probability of falsely authenticating an invalid transmitter. That is, we compare the authentication probabilities (DS 3) between the scenarios where receiver does not

know the key (TS 2) versus when the receiver does know the key (TS 3).

For test scenario 2, the transmitter and receiver are seeded with different keys. The transmitter then sends authenticated packets. The receiver is able to decode the payload because of stealth (Section 5.3.1), but should not accept the packet as authentic because the authentication is not generated using the same key.

Similarly, for test scenario 3, the transmitter and receiver are seeded with identical keys, and the receiver should authenticate the packets.

In our tests, we did not observe any false positives in TS 2, while the authentication performed as usual in TS 3 (as in the robustness tests). We were unable to perform an exhaustive test covering all possible keys for all possible payloads so we cannot conclusively state how secure the authentication system is through this test. However, it does lend some evidence to the security analysis in Chapters 2 and 3.

## 5.4 Conclusion

We have described experimental results obtained via software radios operating at the 2.44GHz center frequency. With our experiment we are able to demonstrate that our scheme is physically realizable and offers good results with low complexity. We observe that the time variation of the channel inhibits the ability of the adversary to distinguish between tagged and untagged signals, especially when the authentication power is low. That is, the hypothesis has low power when the false alarm probability is reasonably low. The results of the experiments detailed above indicate that outside of simulation environments, implementations of this authentication scheme may experience better than expected stealth.

## Chapter 6

### Markov Key Replacement

#### 6.1 Overview of Contributions

In this chapter we propose a new low-complexity solution for the generation, exchange, and replacement of secret keys. We introduce the method in the simplified case where the validity of secret keys are determined without error and prove some helpful properties of randomly generated Markov models. We then extend the method to include non-perfect key recovery and discuss some possible applications.

We make the following contributions:

- We use Markov models in a novel way for key generation and replacement (Section 6.3).
- The proposed method has no explicit signaling costs, low computation complexity, and low storage requirements (Section 6.3.5).
- By using a random access strategy, the method allows the use of very large key spaces without large storage or complexity requirements (Section 6.4.3).
- The proposed method is secure: without the correct model, knowledge of the current key alone gives little information about past and future keys.

Further, knowing the current key gives little information about which model is being used. (Section 6.4.4).

- The proposed method generates keys with a positive entropy rate (Section 6.4.5).

## 6.2 The Key Replacement Problem

In general, secret keys cannot stay secret forever. Whether it is unintentional (e.g., through inadvertent password disclosures) or intentional (e.g., through adversarial attacks), we can never guarantee the secrecy of a key over all time. In fact, the operation of most cryptographic systems disclose some information about the key. When the adversary gains enough key information, she can successfully attack the system, so it is important to replace the key before its lifetime is exceeded.

Consider the authentication problem. When a secret key is used to generate authentication tags, how does Bob know that the tag is genuine? He uses his own secret key to generate a tag which he compares to the observation. If they are identical, then he declares that the tag is genuine. Now if the tag was not based on the secret key, then anyone could generate the tag. In this case, Eve can generate valid tags for arbitrary messages. This is the 'impersonation attack.' Thus the tag needs to demonstrate knowledge of the secret key.

On the other hand, if the tag demonstrates too much information about the secret key, then Eve can eventually recover the key. If she substitutes one of Bob's messages with her own and it is accepted, then she successfully performed a 'substitution attack'. Thus the Eve must have uncertainty about the key.

Maurer [1] formalized the above argument and showed that to prevent im-

personation attacks the authentication tags must contain key information, while to prevent substitution attacks, the authentication also must keep the key secret from the adversary. This indicates the tradeoff between the security offered by the key and its longevity. Since Eve gains key information each time it is used, the key replacement problem is fundamentally important.

### **6.2.1 Example**

Suppose Alice and Bob use secret keys to encrypt the messages that they send to each other. Upon each observation, Eve gains some information about the secret key. Suppose that Eve captures the key after 30 observations of repeated use. Therefore if Alice and Bob need to securely send 100 messages, then they will need between 4 and 100 different keys.

Obviously, it is better to replace keys more often to prevent Eve from gaining key information. However, key replacement does come with some computational or storage requirements, so this is generally a system tradeoff.

### **6.2.2 Key Replacement Paradigms**

Traditional key replacement strategies fall under three paradigms:

1. Distribute secret keys over a secure channel. Generally the secure channel has limited availability.
2. Use third parties to help manage keys. This includes the use of public key infrastructures and key authorities.
3. Negotiate keys over an insecure channel.

We do not assume the continued availability of a secure channel or the existence of helpful third parties. Thus we consider only the last case: the replacement of

keys over insecure channels.

We consider symmetric keys only. It is well known [40] that symmetric keys give the most security per bit, followed by elliptic curve keys, and finally RSA keys give the least security per bit [41][42][43]. That is, for the same length key, a symmetric key system is stronger than asymmetric, public-key based systems. Further, symmetric-key cryptography requires much fewer computations than asymmetric methods.

In the spirit of reducing complexity, we next introduce our key replacement method.

## **6.3 Markov Key Replacement Method**

Suppose that Alice and Bob both share identical Markov models that are in the same state. Let each state of the model correspond to a secret key. We assume that Alice and Bob are synchronized so that their key transitions occur at the same time. For example, they may agree to change their keys at regular time intervals.

We introduce our key replacement method with a simple example.

### **6.3.1 Overview with example**

Alice and Bob agree upon the Markov model shown in Figure 6.1. The four states represent the four possible keys. Alice and Bob are currently using key 2 and their key replacement times are synchronized. At the next key replacement time, Alice starts to use either key 1 or key 3 with equal probability. Suppose Alice chooses key 1.

Bob, having both the model and the current key, knows that the possible

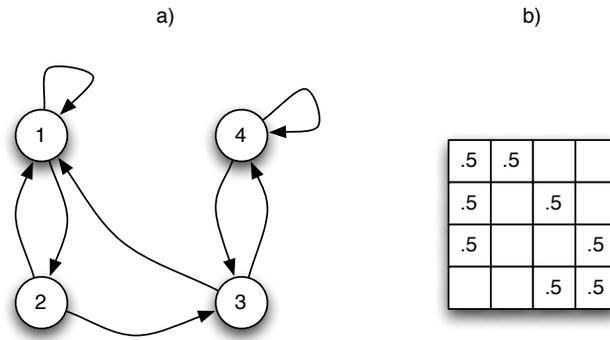


Figure 6.1: Simple Markov Model when  $K = 4$  and  $b = 2$ . The states and possible transitions are shown in (a) while a corresponding probability matrix is shown in (b). Note that  $\mathbf{A}$  is sparse.

keys are 1 and 3. He is able to determine which key is correct by checking the message integrity (detailed below) for each key. After the check, Bob knows that Alice is using key 1, and starts to use key 1.

Alice and Bob thus regain synchrony. At the next key transition, the above steps are repeated.

### 6.3.2 Message model

We adopt the following message model, where we use the superscripts  $a, b$  to denote the influence of Alice and Bob, respectively.

At time  $i$ , Alice uses her secret key  $k_i^a$  to encrypt message  $s_i^a$  to form the ciphertext  $x_i$

$$x_i = f_e(s_i^a, k_i^a), \tag{6.1}$$

and Bob decrypts the ciphertext with his key  $k_i^b$

$$s_i^b = f_d(x_i, k_i^b). \tag{6.2}$$

The encryption and decryption functions satisfy

$$s = f_d(f_e(s, k), k) \tag{6.3}$$

We assume that the message includes an integrity check (e.g., checksum) so that the receiver knows when the message is received correctly ( $s_i^b = s_i^a$ ). For example, the message may be appended with a CRC-32 (cyclic redundancy check) which will only match when the correct key is used to decrypt. Writing the integrity check as  $\psi(\cdot)$ , we have

$$\psi(x_i, k) = \begin{cases} 0 & \implies s_i^a \neq s_i^b \text{ or } k_i^a \neq k \\ 1 & \implies s_i^a = s_i^b \text{ and } k_i^a = k \end{cases} . \tag{6.4}$$

That is, when the integrity check fails Bob is certain that the message is received in error or that the keys are mismatched. For clarity of discussion, we have assumed that a passed integrity check implies perfect message reception with the correct key. However, in actuality there may be a non-zero probability that an incorrect key or message can lead to a passed integrity check. We elaborate on the effect of errors in Section 6.3.6. Therefore Bob can use the outcome of  $\psi(\cdot)$  to determine if he is using the correct key.

### 6.3.3 Markov model

Denote a Markov model by  $\lambda = (\pi, \mathbf{A})$ , where  $\pi$  contains the initial key probabilities and  $\mathbf{A}$  is the transition probability matrix.

Let the size of the keyspace be  $K$ . Thus  $\pi$  is a  $K \times 1$  vector where  $\pi(m)$  is the probability that the initial key is  $m$ . Likewise,  $\mathbf{A}$  is a  $K \times K$  matrix where  $A(m, n)$  is the probability that the next key will be  $n$  given that the current key is  $m$ .

Suppose that there exists a large codebook of Markov models  $\Lambda$ , where the size of the codebook is  $|\Lambda| = L$ . Let Alice and Bob share a secret index  $0 \leq l < L$  which they use to select the same model  $\lambda^a = \lambda^b = \Lambda(l) = (\pi, \mathbf{A})$ .

Denote the keys that Alice uses by  $k_0^a, k_1^a, k_2^a, \dots$ . Similarly, Bob's keys are denoted by  $k_0^b, k_1^b, k_2^b, \dots$ . Here the subscript indicates key epochs, so that  $k_0^a$  is used in the epoch 0 and  $k_1^a$  is used in epoch 1. We assume that the models are synchronized so that Alice and Bob transition between keys at the same time. They use their agreed-upon Markov models  $\lambda^a = \lambda^b$  to transition between keys; the resulting key sequences are Markov chains.

For synchronization purposes, we assume that  $\pi$  deterministically selects the initial key so that  $(\lambda^a = \lambda^b) \implies (k_0^a = k_0^b)$ . We further assume for complexity purposes that each key can transition to exactly  $d$  other keys. We call  $d$  the *branching factor* of the model, and we specify that the key transitions are equiprobable, i.e., if a transition can occur between keys  $m$  and  $n$  it occurs with probability  $1/d$ .

**Definition 1** *The keys of Alice and Bob are synchronized for  $n$  key epochs when*

$$k_i^a = k_i^b, 0 \leq i < n \quad (6.5)$$

*The keys lose synchrony at time  $n_0$  for*

$$n_0 = \arg \min_i k_i^a \neq k_i^b \quad (6.6)$$

Clearly,  $n > 0$  since the initial keys are the same. We wish to assert that  $n_0$ , the time-to-failure, is large. It is simple to show that  $n_0 = \infty$  when the integrity check is perfect (Section 6.3.4).

### 6.3.4 Key Replacement Algorithm

We detail the simple key replacement algorithms below. First we consider the actions of the sender and then the receiver.

#### Key Replacement (Alice)

At the key replacement time, Alice uses  $\lambda^a$  and  $k_i^a$  to generate the next key  $k_{i+1}^a$ :

1. Find the set of possible keys from  $k_i^a$ :  $\mathcal{N} = \{n | A(k_i^b, n) > 0\}$ .
2. Assign  $k_{i+1}^a = n \in \mathcal{N}$  w.p.  $1/d$

After the key is chosen, Alice uses it to encrypt future messages. She does not explicitly signal any key information to Bob.

#### Key Recovery (Bob)

At the key replacement time, Bob starts to receive messages encrypted with a different key. Assume that  $k_i^a = k_i^b$ . Since  $\lambda^b = \lambda^a$ , Bob knows the transition probabilities to the next key.

1. Find the set of possible keys from  $k_i^b$ :  $\mathcal{N} = \{n | A(k_i^b, n) > 0\}$ .
2. For each key  $n \in \mathcal{N}$ :
  - If  $\psi(x_i, n) = 1$  then assign  $k_{i+1}^b = n$  and halt.
  - Else continue.

Since we assume that  $k_i^a = k_i^b$  and  $\lambda^b = \lambda^a$ , Bob considers the same keys as Alice does during her key replacement. Since we assume that the integrity check is perfect (equation (6.4)), Bob is guaranteed to select the correct key.

Given that Alice and Bob share the same model, they are initially synchronized with the same first key. From above, it is a simple inductive step to see that Bob will remain in key synchrony with Alice. Thus the time-to-failure  $n_0 = \infty$ , i.e., they will never lose key synchrony.

### 6.3.5 Complexity and Costs

We now consider the communication, memory, and computation requirements of this scheme.

Aside from the initial synchronization of Markov models, there is no further explicit communications between Alice and Bob. That is, after sharing  $\log_2(L)$  bits of information for initialization, the method requires no additional bits to be transmitted over the channel.

Both the transmitter and the receiver need to have in memory the set of possible future keys. Since there are  $d$  possible keys, the minimum storage requirement is exactly  $d \log_2(K)$  bits. We will show in Section 6.4.1 that this is achievable - it is not necessary (or possible in most instances) to store the entire set of Markov models in memory.

In terms of computation, the transmitter only needs to select a key from the set of possible future keys. The receiver, however, needs to check the set of keys until the correct one is found. Thus he needs to perform between 1 and  $d$  decryptions and integrity checks. Since the next key is chosen with equal probability from the set, the receiver will perform  $d/2$  decryptions and integrity checks on average.

### 6.3.6 Imperfect Key Recovery

In the previous section we assumed that Bob is able to determine the correct key without error. This is usually a good approximation, but in this section we determine the effects of detection error on the proposed key replacement method.

Suppose that the ciphertext  $x_i$  has been encrypted with key  $k_i$ :

$$x = f_e(s, k_i) \quad (6.7)$$

When we test a single key, there are 2 cases where the key recovery fails.

1. Missed detection: the correct key is tested ( $k = k_i$ ) but fails the check with probability  $1 - p > 0$ :

$$\psi(x, k = k_i) = \begin{cases} 0 \text{ w.p. } 1 - p \\ 1 \text{ w.p. } p \end{cases} \quad (6.8)$$

$$(6.9)$$

2. False alarm: the wrong key is tested ( $k \neq k_i$ ) but passes the check with probability  $\alpha > 0$ :

$$\psi(x, k \neq k_i) = \begin{cases} 0 \text{ w.p. } 1 - \alpha \\ 1 \text{ w.p. } \alpha \end{cases} \quad (6.10)$$

Recall that the recovery performed by Bob (Section 6.3.4) checks the validity of  $d$  keys. We assume that the trials have independent outcomes. Considering a set of  $d$  keys, there are four possible outcomes.

1. The correct key passes the integrity check (all others fail). This occurs with probability

$$Pr[\text{Case 1}] = p(1 - \alpha)^{d-1} \quad (6.11)$$

2. The wrong key passes the integrity check (all others fail). This occurs with probability

$$Pr[\text{Case 2}] = (d - 1)(1 - p)\alpha(1 - \alpha)^{d-2} \quad (6.12)$$

3. No keys pass the integrity check. This occurs with probability

$$Pr[\text{Case 3}] = (1 - p)(1 - \alpha)^{d-1} \quad (6.13)$$

4. Multiple keys pass the integrity check. This occurs with probability

$$Pr[\text{Case 4}] = 1 - Pr[\text{Case 1, 2, or 3}] \quad (6.14)$$

Looking ahead to Section 6.3.6, the probability of a correct detection  $Pr[\text{Case 1}]$  needs to be sufficiently high, otherwise Alice and Bob will lose synchrony quickly. In the following section we modify the recovery algorithm to improve the detection probability in the midst of errors.

### Extended Recovery Algorithm

In general, a key is used for multiple messages. Rather than determining the replacement key after a single message, consider the utility of using  $C$  messages. For the key epoch  $i$ , we denote the messages using the same key as  $x_i^1, x_i^2, \dots$

The number of times the check passes is independent for each key. Bob tallies the number of times a key passes the check over the  $C$  messages, and selects the key that has the most positives.

Again assume that  $k_i^a = k_i^b$  and  $\lambda^b = \lambda^a$  so that Bob knows the transition probabilities to the next key.

1. Find the set of possible keys from  $k_i^b$ :  $\mathcal{N} = \{n | A(k_i^b, n) > 0\}$ .

2. For each key  $n \in \mathcal{N}$ :

$$T[n] = \sum_{c=1}^C \psi(x_i^c, n) \quad (6.15)$$

3. Select  $k_{i+1}^b = \arg \max_n T[n]$

The probability that the correct key will be detected  $c$  times out of  $C$  is given by the binomial probability  $B(c, C, p)$ . Similarly, for the incorrect keys the probability is  $B(c, C, \alpha)$ . Therefore since the trials have independent outcomes, the probability that the Bob chooses the correct key is the probability that the correct key is detected more times than any other key.

$$\tilde{p} = Pr[k_{i+1}^b = k_{i+1}^a] \quad (6.16)$$

$$= \sum_{c=1}^C B(c, C, p) \left( \sum_{d=0}^{c-1} B(d, C, \alpha) \right)^{d-1} \quad (6.17)$$

Aside from the detection and false alarm probabilities ( $p, \alpha$ , respectively), the number of messages  $C$  determines the detection probability  $\tilde{p}$  of the correct key. Figure 6.2 shows that the probability of choosing the wrong key falls exponentially as  $C$  increases. Further, the test does not need to consider large  $C$  when the detection probability  $p$  is high.

### Synchronization Performance

The probability that Bob chooses the incorrect key is  $\varepsilon = 1 - \tilde{p}$ . Thus the probability that Alice and Bob lose synchrony at epoch  $n_0$  is

$$Pr(\text{Lost at } n_0) = \varepsilon(1 - \varepsilon)^{n_0-1} \quad (6.18)$$

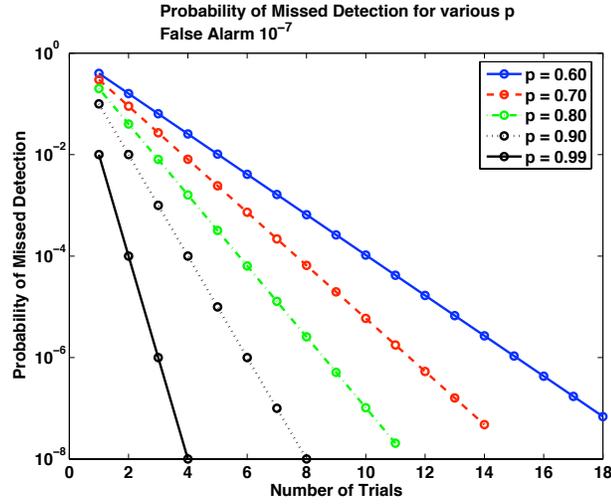


Figure 6.2: Probability of choosing the incorrect key for various raw detection key probabilities  $p$  when  $\alpha = 10^{-7}$ . Extending the key replacement over multiple messages encrypted with the same key improves the detection probability. Lower  $p$  require more observations for good detection probability.

We are interested in the case when Alice and Bob maintain synchrony for at least  $n_0$  with a certain probability. That is,

$$Pr(n_0 > n) = \sum_{n < n_0} Pr(\text{Lost at } n_0) \quad (6.19)$$

$$= 1 - (1 - \varepsilon)^{n-1} \quad (6.20)$$

$$n = \frac{\log(1 - Pr(n_0 > n))}{\log(1 - \varepsilon)} \quad (6.21)$$

For example, we calculate that when  $\varepsilon = 10^{-4}$ , then with probability 99.99% Alice and Bob remain synchronized for at least  $9.2 \times 10^4$  keys. Figure 6.3 shows the number of key transitions before failure for various confidence levels.

## Complexity and Costs

When the recovery of the key is extended for  $C$  messages, the number of computations and memory requirements necessarily increase.

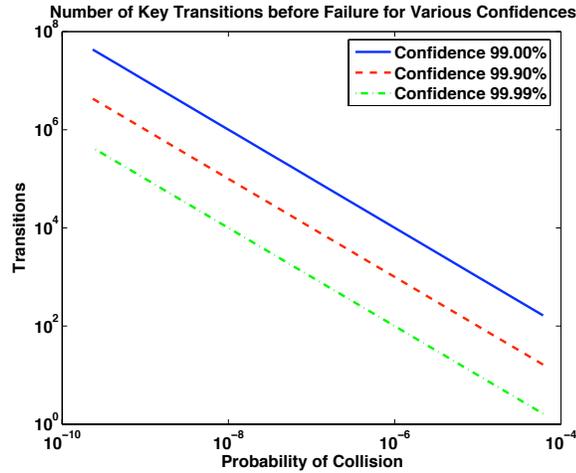


Figure 6.3: Number of key transitions before failure for various confidences.

Since the computations are identical for each message, the computations increase by a factor of  $C$ . The final step of selecting the key with the largest score requires negligible computation.

As for the storage requirements, the set of possible keys remains constant over the duration of the test, so there is no additional cost. However, there is the need to keep the tally of how many times each key passes the integrity checks. This requires  $\log_2(C)$  bits per key, or  $d \log_2(C)$  bits in total. Note that there was no need to keep score in the previous section because exactly one key would satisfy the integrity check.

## 6.4 Markov Model

### 6.4.1 Specification

For synchronization purposes, the choice of model  $\lambda$  determines the initial key  $k_0$ . That is,  $\pi$  is non-zero for exactly one entry.

To limit the complexity of the receiver (Section 6.3.4), we focus on the case

	32 bits	32 bits
key 0	next key	next key
key 1	next key	next key
...	...	...
key K-1	next key	next key

Figure 6.4: A fully specified Markov Model with 32-bit keys,  $K = 2^{32}$ , branching factor  $d = 2$ . Requires at least 1 gigabyte to store.

where the branching factor of the model is fixed to be exactly  $d$  for all keys. That is, each row of  $\mathbf{A}$  has exactly  $d$  non-zero entries. The assumptions on  $\pi$  and  $\mathbf{A}$  may be written

$$\sum_m I(\pi(m)) = 1 \quad (6.22)$$

$$\forall m, \sum_n I(A(m,n)) = d \quad (6.23)$$

where  $I(\cdot)$  is the indicator function.

### 6.4.2 Construction

How should the Markov models be specified? The Markov model may be fully specified and stored in memory. The storage requirements for a single model are  $O(dK)$ . This quickly becomes infeasible when  $L$  or  $K$  grow large. For example, a single Markov model for (unrealistically small) 32 bit keys would require at least 512 megabytes ( $2^{32}$  bits = 512 megabytes).

The need for multiple models increases the storage requirements to  $O(dKL)$ . With  $L = K = 2^{32}$  and a modest branching factor of  $d = 2$ , the storage requirement is approximately  $2^{32+32+1}$ , or 4.3 billion gigabytes (1 gigabyte =  $2^{33}$  bits). Clearly, with even modest key sizes the storage requirements can be prohibitive.

We therefore turn our attention towards a randomly accessible Markov model.

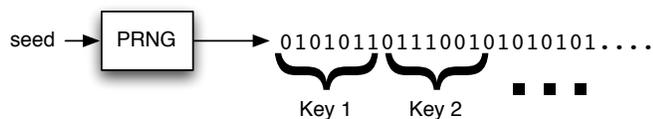


Figure 6.5: Construction of a random access Markov Model.

That is, we do not store the entire Markov model in memory, but we are able to directly query the transition probabilities for a given key.

We assume  $m$ -bit keys. How do we find the state transitions from key  $i$ ? We outline one possibility below.

1. Seed the pseudo-random number generator (PRNG) with a value  $f(l, i)$ , where  $l$  is the key that chooses the model  $\lambda$  and  $i$  is the current key. For example,  $f(\cdot)$  may be  $f(l, i) = K * l + i$
2. Until there are  $d$  unique keys, select  $m$ -bit chunks of the PRNG output. Each chunk corresponds to a candidate next key.
3. The transition probabilities of each candidate key are  $1/d$ .

This process is diagrammed in Figure 6.5. Upon completion of these steps, the transition probabilities for the current state are specified.

Since  $\pi$  chooses a single initial key, it is easy to specify. One possibility is to always choose the first key reachable from key 0.

### 6.4.3 Model Properties

Consider an arbitrary model  $\lambda = \lambda^a = \lambda^b$  constructed as in Section 6.4.2 and a corresponding sample key sequence  $k_0^a, k_1^a, \dots$ . What are the desirable properties for such a sequence?

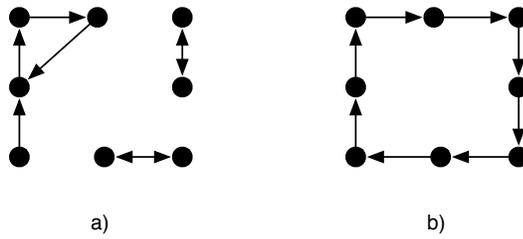


Figure 6.6: a) Small reachable subspace b) Periodic

First, the keys should be drawn from a large space. If the space were small, the adversary could successfully attack with non-negligible probability. As a simple example, with a space of only 4 keys, an adversary correctly guesses the key with probability  $1/4$ . Thus  $K$  should be large.

Closely related is the reachable subspace of  $K$  given any initial key  $k_0$ . A key  $v$  is reachable from  $u$  if there exists a path from  $u$  to  $v$ :  $u \rightsquigarrow v$ . If it can reach many keys, then it is difficult for the adversary to recover the key information. As a simple example, consider the graph in Figure 6.6a. Given any key, the reachable space is small relative to the size of the entire keyspace. This is clearly suboptimal.

Finally, the graph should not be periodic. In Figure 6.6b the entire space is reachable given any initial key, but the regular nature of the key transitions makes it easier for the adversary to guess the next key. Note that in Figure 6.6a the situation is even more dire because the periods are small.

Therefore we have three desirable properties for the models:

1.  $K$  is large
2. The reachable subspace of  $K$  is large for almost all keys
3. The resulting key sequences are aperiodic

We show that satisfaction of the above properties depends on the keyspace  $K$

and the branching factor  $d$ .

### **$K$ is large**

The size of the keyspace  $K$  plays an important role in securing the secret key. When  $K$  is small, it is possible for the adversary to test all possible keys and recover the correct key (e.g., by using equation (6.4)). This is known as the brute-force method since no intelligence is needed, only raw computation ability.

Therefore  $K$  should be large so that the adversary cannot use brute-force methods to overcome the system. However, the birthday paradox states that  $O(\sqrt{K})$  computations are usually sufficient to find a collision between two ciphertexts given distinct inputs. Therefore the condition is that  $\sqrt{K}$  is sufficiently large so that it is infeasible for the adversary to perform a brute-force attack.

### **Reachable Keyspace**

We show that given any initial key, the reachable keyspace is large, that is,  $O(K)$ . First we introduce the concept of strongly connected components. We show that the existence and size of a single giant connected component (GCC) depends on the branching factor  $d$ . Finally, we show that with probability 1, a) all keys can reach the GCC and b) no keys leave the GCC.

#### **Strongly Connected Components**

As specified in Section 6.4.1, the Markov models are random directed graphs (digraphs) with constant out-degree. A digraph  $\mathcal{S}$  is a *strongly connected* when there exists a path between any randomly chosen pair of vertices  $u, v \in \mathcal{S}$ :

$$u \rightsquigarrow v \tag{6.24}$$

$$v \rightsquigarrow u \tag{6.25}$$

The strongly connect components (SCCs) of a digraph are the maximal strongly connected subgraphs. The interpretation in our case is the following. If the current key is in an SCC, any other key in the same SCC may be chosen in the future with non-zero probability.

### **Size of the GCC**

When the size of a SCC reaches  $O(K)$ , it is typically referred to as the giant connected component (GCC) since it dominates the other SCCs in size. For a given digraph, the SCCs may be identified using efficient algorithms such as Tarjan’s algorithm [44] or Gabow’s algorithm [45].

When the graphs are constructed randomly as in Section 6.4.2, we give the size of the GCC in probability. We outline the procedure and then give the results.

First we define the concept of fan-out and fan-in. The fan-in of a node  $v$  is the set of vertices  $u$  for which there exists a path  $u \rightsquigarrow v$ . Similarly, the fan-out of a node  $u$  is the set of nodes  $v$  for which there exists a path  $u \rightsquigarrow v$ . The fan-out (or fan-in) of a vertex is large when its size is  $O(K)$ . Let  $L^+$  be the set of vertices with a large fan-out, and let  $L^-$  be the set of vertices with a large fan-in.

Intuitively, when a node  $u$  has a large fan-out and a distinct node  $v \neq u$  has a large fan-in, there exists a path  $u \rightsquigarrow v$  with high probability. Thus when nodes have both large fan-in and large fan-out, they are connected with high probability. The above statements are made precise in [46]. We highlight the relevant results below.

Let  $\pi^-$  (resp.,  $\pi^+$ ) be the probability that a randomly chosen vertex has a large fan-in (resp., large fan-out). It follows that  $|L^-| = \pi^- K$  and  $|L^+| = \pi^+ K$ .

They are the smallest non-negative solutions of

$$1 - \pi^- = \sum_i p_i^- (1 - \pi^-)^i \quad (6.26)$$

$$1 - \pi^+ = \sum_i p_i^+ (1 - \pi^+)^i \quad (6.27)$$

where  $p_i^-$  (resp.,  $\pi^+$ ) is the probability that a key has exactly  $i$  incoming (resp., outgoing) transitions.

$$p_i^- = \binom{K}{i} p^i (1-p)^{K-i} \quad (6.28)$$

$$p_i^+ = \begin{cases} 1 & i = d \\ 0 & \text{otherwise} \end{cases} \quad (6.29)$$

Note that each vertex has constant out-degree  $d$  but variable in-degree. In our scenario, the probability that there exists a transition between a randomly chosen pair of source and destination keys is  $p = d/K$ , and so the in-degree distribution is given by the binomial probability mass function with parameter  $p$ .

When  $d > 1$ ,  $\pi^-$  has a unique solution in  $(0, 1)$ , and  $\pi^+ = 1$ . In other words, a positive fraction of vertices have large fan-in while all vertices have large fan-out. Based on the argument above, the size of the GCC is approximately

$$|\mathcal{G}| \cong |L^- \cap L^+| \quad (6.30)$$

$$\cong |L^-| + |L^+| - |L^- \cup L^+| \quad (6.31)$$

$$\cong (\pi^+ + \pi^- - (1 - \psi)) K \quad (6.32)$$

where

$$\psi = \sum_{i,j} p_{ij} (1 - \pi^-)^i (1 - \pi^+)^j \quad (6.33)$$

$$= 0 \quad (6.34)$$

since  $\pi^+ = 1$ . Thus, the GCC  $\mathcal{G}$  is unique with size

$$|\mathcal{G}| \cong \pi^- K \quad (6.35)$$

with probability approaching 1 as  $K \uparrow \infty$ .

Table 6.1 shows the size of the GCC in theory and in practice for various branching factors. For each value of  $d$ , 100 matrices of size  $K = 2^{10}$  were generated. The size of the GCC was found using Tarjan's algorithm and averaged over each realization. Shown in the table is the ratio of GCC size to  $K$ . The theoretical GCC size matches very well with the empirical evidence. Clearly, increasing  $d$  increases the proportion of the GCC, though the gains diminish after  $d = 4$ .

When the size of the GCC is less than  $K$ , the size of the reachable subspace is diminished. In bits, we say that the penalty is

$$\text{Penalty} = -\log 2(|\mathcal{G}|/K) \tag{6.36}$$

For example, if the GCC has size  $2^9$  but the keyspace has size  $2^{10}$ , then the penalty is 1 bit. Thus the keys that are traversed offer the security of a 9-bit key instead of a 10-bit key.

Note the significant decrease in penalty as the branching factor is increased from 2 to 4. We observe diminishing improvements for larger branching factors.

### **Reachability of GCC**

Since the GCC  $\mathcal{G}$  is large, any key  $k \in \mathcal{G}$  has a reachable subspace with size  $O(K)$  by definition. However, what about those keys that are not in  $\mathcal{G}$ ? We need to show that with high probability, a) the initial key will enter  $\mathcal{G}$ , and b) given that a selected key is in the GCC, future keys will remain in the GCC.

First observe that when a key leaves the GCC, it cannot come back (it has no path). Suppose that the key is  $v \notin \mathcal{K}$ . If there existed a path from  $v \rightsquigarrow \mathcal{K}$ , then since  $\mathcal{K} \rightsquigarrow v$  this implies that  $v \in \mathcal{K}$ , which is a contradiction.

Consider the keys that are connected to  $\mathcal{G}$ . Let  $\mathcal{G}^-$  be the set of keys that

Table 6.1: Size of the GCC

Branching factor	GCC Proportion		Penalty (bits)
	Theory	Empirical	
2	.7972	.7972	.3369
3	.9408	.9404	.0881
4	.9803	.9801	.0287
5	.9931	.9932	.0100
6	.9975	.9977	.0036
7	.9991	.9990	.0013
8	.9997	.9997	.0005

have a path to  $\mathcal{G}$  and let  $\mathcal{G}^+$  be the set of keys that are reachable from  $\mathcal{G}$ .

$$\mathcal{G}^- = \{u | u \rightsquigarrow \mathcal{G}\} \quad (6.37)$$

$$\mathcal{G}^+ = \{v | \mathcal{G} \rightsquigarrow v\} \quad (6.38)$$

The resulting set  $\mathcal{B} = \mathcal{G}^- \cup \mathcal{G} \cup \mathcal{G}^+$  forms a subgraph that is termed a *bowtie digraph* as shown in Figure 6.7. The wings of the bowtie are formed by two wings:  $\mathcal{G}^+$  with vertex set  $L^+ \cap \overline{L^-}$  and  $\mathcal{G}^-$  with vertex set  $\overline{L^+} \cap L^-$ . Thus  $\mathcal{B} = L^+ \cup L^-$ .

Recall that  $|L^+| = K$  and hence  $L^+ = \mathcal{K}$ . Hence  $\mathcal{B} = \mathcal{K}$ , i.e., the bowtie graph encompasses all vertices in the space. By the same fact, we also have that  $\overline{L^+} = \emptyset$  and therefore  $\mathcal{G}^- = \emptyset$ . Therefore, any randomly chosen key is either in  $\mathcal{G}^+$  or  $\mathcal{G}$ .

The structure of  $\mathcal{B}$  yields the following properties:

1. If a key is not in  $\mathcal{G}$ , then it is in  $\mathcal{G}^+$ . Therefore, with probability 1 the

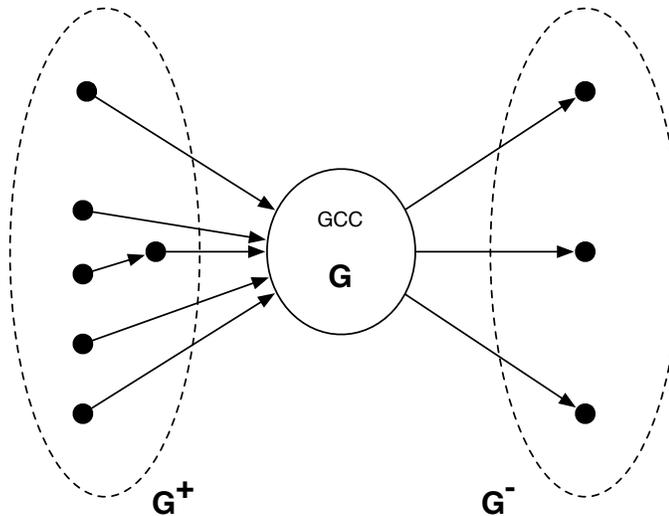


Figure 6.7: An example bowtie digraph connected to the giant connected component  $\mathcal{G}$ .  $\mathcal{G}^+$  leads into  $\mathcal{G}$ ,  $\mathcal{G}^-$  emanates from  $\mathcal{G}$ .

current state will eventually be in  $\mathcal{G}$ . With probability approaching 1 as  $K \uparrow \infty$ , any randomly chosen key will enter the GCC.

2. Once a key is in the GCC, it will not depart since with probability approaching 1 as  $K \uparrow \infty$ ,  $|\mathcal{G}^-| = 0$ .

### Aperiodicity of GCC

A strongly connected graph is periodic if the GCD of all cycle lengths is strictly  $> 1$ . Random digraphs with constant branching factor are aperiodic with high probability. We can loosely bound the probability of having a periodic GCC by reasoning as follows:

1. Suppose that it is possible to remove transitions from the GCC until it is a minimally connected periodic graph. That is, removing one more transition will make the graph neither connected nor periodic. Suppose that the resulting subgraph has period  $x > 1$ .

2. Since the subgraph is periodic, we can color the nodes so that only like-colored nodes are reachable in length  $nx$  transitions, where  $n$  is any positive integer. By periodicity, any path through the subgraph has the same sequence of colors, modulo rotations. For example, if  $x = 2$  then the color sequence is always black-white-black-white etc.
3. In order for the GCC to be periodic, all the transitions must follow the color sequence of the periodic subgraph. Since the transitions are randomly generated, the probability of this is

$$\left(\frac{1}{x}\right)^{(d-1)O(K)} \tag{6.39}$$

which is vanishing small for large  $K$  and  $x > 1$ . Therefore, with high probability, the GCC is aperiodic.

4. Since in step 1 we assumed the a periodic subgraph existed, we have found an upper bound.

Therefore with reasonably large  $K$ , the probability that a randomly generated digraph with constant out-degree is aperiodic is very high.

#### 6.4.4 Codebook Properties

We show that the following properties about the codebook  $\Lambda$  hold:

1.  $H(k_{i+1}|k_i) \cong H(\mathcal{K})$  (Entropy of next key)
2.  $H(l|k_i) \cong H(\Lambda)$  (Entropy of model in use)

where  $\mathcal{K}$  is the keyspace and  $\Lambda$  is the set of Markov models. That is, given the current key, little information is revealed about the next key or which Markov model is currently being used.

## Entropy of Next Key

Given a model and the current key, there are  $d$  candidates for the next key. Over all  $L$  models, there are therefore  $dL$  candidates uniformly distributed over the  $K$  possible keys. The probability of transitioning from key  $m$  to key  $n$  is therefore

$$\frac{1}{dL} \sum_l A^l(m, n) = \frac{1}{dL} |A(m, n)| \quad (6.40)$$

Let

$$(6.41)$$

The entropy is therefore

$$H(k_{i+1}|k_i = m) = \sum_n h\left(\frac{1}{dL} |A(m, n)|\right) \quad (6.42)$$

$$\text{where } h(x) = -x \log_2(x) \quad (6.43)$$

Since the keys are chosen uniformly over each  $A$ ,  $P(|A(m, n)| = x)$  is approximated by the Poisson distribution

$$P(|A(m, n)| = x) = f(x, \bar{x}) \quad (6.44)$$

$$= \frac{\bar{x}^x e^{-\bar{x}}}{x!} \quad (6.45)$$

where  $\bar{x}$  is the expected number of occurrences. In our case,  $\bar{x} = dL/K$ .

The entropy of the next key can thus be approximated

$$H(k_{i+1}|k_i = m) = \sum_x E[|A(m, n) = x|] h\left(\frac{x}{dL}\right) \quad (6.46)$$

$$= \sum_x K f(x, \bar{x}) h\left(\frac{x}{dL}\right) \quad (6.47)$$

Figure 6.8 shows that the entropy is close to the maximum when the branching probability is high and the number of models  $L$  is high. The critical point is that  $dL$  must be greater than  $K$ ; this ensures that over all the models the possible keys are sufficiently random. Note that with sufficiently many models, the branching factor contributes only a small amount of entropy.

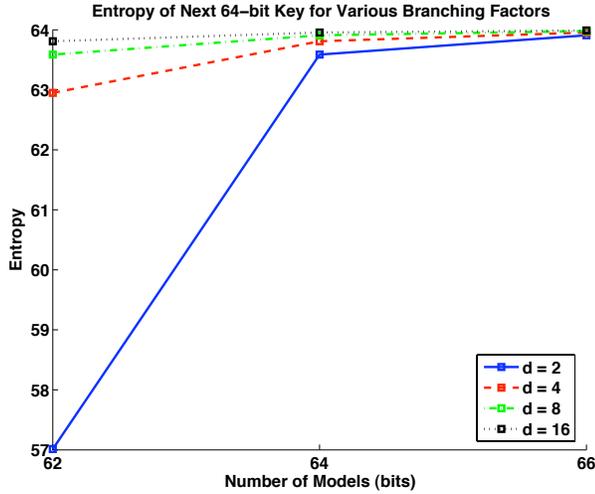


Figure 6.8: Entropy of next key.  $K = 2^{64}$ .

### Entropy of Model in Use

The current state should give information about which model  $\lambda$  is being used. From Section 6.4.3 we know that the GCC  $\mathcal{G}$  has size  $O(K)$  and that the number of non-zero stationary probabilities  $\mu_i$  are also  $O(K)$ . For large  $d$ , we can approximate  $|\mathcal{G}| \cong K$ .

Assume that the Markov chain is in steady state. Let the stationary probability of key  $i$  under model  $l$  be  $\mu_i^l$ . Then the entropy of the model given the key is

$$H(l|k_i) = \sum_l h\left(\frac{\mu_i^l}{\sum_l \mu_i^l}\right) \quad (6.48)$$

We note from experiment that the stationary probabilities are well approximated by a Rayleigh distribution when  $d$  is small and a normal distribution when  $d$  is large (Figures 6.9 and 6.10 respectively). Thus when there are  $L$  independent instances of  $\mu_i^l$  they will follow this distribution.

Figure 6.11 shows that the model entropy is close to the maximum when the branching probability is high and the number of models  $L$  is high. Increasing the

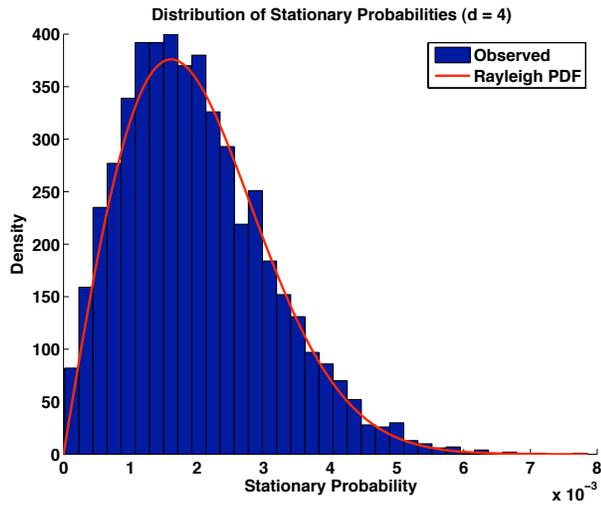


Figure 6.9: Distribution of stationary probabilities are well approximated with Rayleigh distribution for small  $d$ .

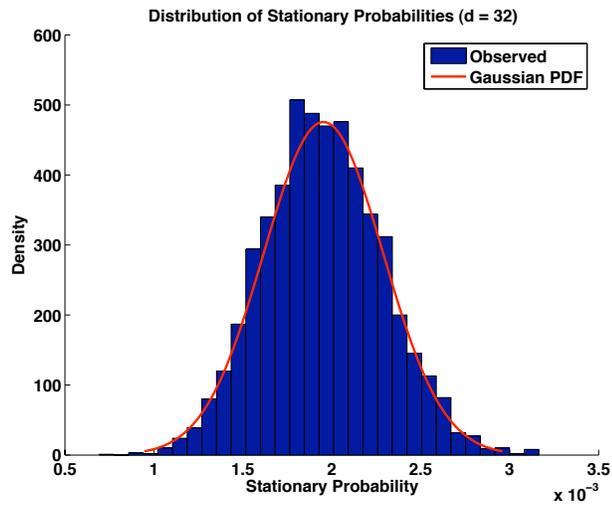


Figure 6.10: Distribution of stationary probabilities are well approximated with Gaussian distribution for large  $d$ .

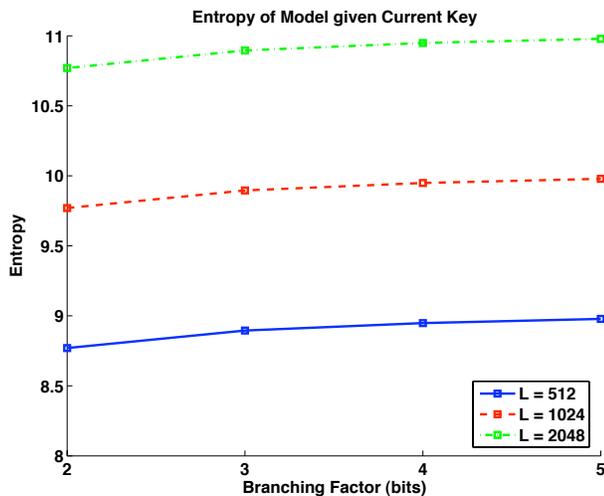


Figure 6.11: Entropy of Model given current key.  $K = 2^{10}$ .

branching factor improves the model entropy.

### 6.4.5 Entropy Rate

Since the GCC is strongly connected, it is irreducible. From Section 6.4.3 it is also aperiodic. An irreducible and aperiodic Markov chain converges to its unique stationary distribution  $\mu$ , and the resulting entropy rate is

$$H(\mathcal{X}) = - \sum_{m,n \in \mathcal{G}} \mu_m A_{mn} \log A_{mn} \leq \log d \quad (6.49)$$

with equality when a transition between keys  $m$  and  $n$  implies that  $\mathbf{A}_{mn} = 1/d$ . Thus entropy rate is maximized when future keys are equiprobable.

Note that the calculated entropy rate is for a given Markov model. Given that it is known, the entropy rate is given by equation (6.49). When the model is not known, the entropy of the next key was shown to be high, near the entropy of the keyspace  $\mathcal{K}$  (Section 6.4.4).

Having a higher entropy rate is beneficial for security since a model will become associated with many paths rather than only a few. (Consider the multi-

plicity of paths that arise when  $d$  is increased from 1.) This in turn makes it more difficult for the adversary to determine which model is being used. She may try to use the Baum-Welch algorithm[47] but it is not accurate unless the number of observations is very high with respect to size of the state space. Of course, with large keys will not be the case.

## 6.5 Applications

We give a few examples of how this key replacement method may be used. This list is by no means comprehensive and is meant to give a taste of the possibilities.

### 6.5.1 Cryptography

The proposed method may be used to replace keys in cryptographic frameworks. In cryptography, all data assumed to be received without error. That is, the physical layer is abstracted away to provide an error-free channel.

In this case, the correct key always passes the integrity check, so  $p = 1$ . The probability that an incorrect key passes the check is given by the collision probability  $\alpha$ . For example, when the integrity check is a 32-bit tag, the probability that two randomly chosen unique keys have the same tag are  $2^{-32} \cong 2 \times 10^{-10}$ . Thus a single message is sufficient to determine the next key with high probability, and the number of transitions to failure is very high (Section 6.3.6).

### 6.5.2 Physical Layer Authentication

The proposed method may be used to replace the secret keys in the physical layer authentication framework (Chapters 2 and 3). Here, the secret key does not effect the recovery of the messages from Alice, but effects the ability of Bob

to authenticate Alice.

In this case, the detection probability  $p$  is never unity due to the random channel and noise. However, the false alarm probability can be set arbitrarily low as a design parameter. As shown previously, channels with low SNR do not offer very high detection probability, and thus the modified recovery algorithm of Section 6.3.6 must be used in order to keep Alice and Bob in synchrony.

However, there is an interesting tradeoff between  $p$  and the authentication probability. When the next key is not immediately recovered, Bob must delay his decision of Alice's authenticity because it is dependent on having the correct key. For those messages, the decision of authenticity are made retroactively and are delayed for up to  $C$  message periods.

### 6.5.3 Frequency Hopping Communications

Rather than restricting ourselves to keys, we consider the application of this method to another situation where a variable changes pseudo-randomly. In frequency-hopped systems, data is transmitted over different carriers that change at deterministic intervals. Rather than having the data carriers change deterministically as well, the proposed method may be used to select the carrier (or set of carriers). Of particular interest are multi-carrier authentication systems where the placement of the authentication tag may be chosen according to this method.

Suppose for simplicity that the system uses one carrier at a time. The detection and false alarm probabilities are determined by the effective SNR of the channel. When the carrier changes, the receiver can use an energy detector to scan the set of  $d$  possible next carriers and decide which contains a signal and which contains noise only.

The analogue of using multiple messages in the previous situations is to extend

the energy detection over time to provide a better estimate of which carrier is being used. Also, the increased complexity lies in the necessity of monitoring  $d$  carriers simultaneously. However, this may be less of an issue in software-based radio where much of the computation is done in software and not hardware.

## 6.6 Related Work

Without assuming the use of secure channels or third parties, Alice and Bob must negotiate keys over the insecure channel. Public key (asymmetric) cryptography (e.g., Diffie-Hellman algorithm [48]) is often used for this purpose. However, it is computationally intensive since it relies on operations such as modular exponentiation. In comparison, symmetric key cryptography is fast and efficient but does not lend itself to key negotiation.

For this reason, many systems use public key cryptography to exchange symmetric keys which are used to encrypt the bulk of the communications. When it is time to replace the key, Alice and Bob can restart the key negotiation to create a fresh key.

There has been work towards reducing the cost of key exchange after the initial setup. One of the simplest schemes is for Alice use the current key  $k_i$  to encrypt the future key  $k_{i+1}$  and sent it to Bob. This has the benefit of avoiding the additional cost of key negotiation. However, in the event that Eve does acquire a key, all subsequent keys are made accessible to her.

The idea of hash chains [49] defends against this scenario. Suppose that Alice and Bob agree upon a key  $k_0$  and an index  $n$ . Then given  $H(\cdot)$ , they calculate the keys  $k_j = H^j(k_0)$  where  $H^j(\cdot)$  is a one-way function applied  $j$  times to the input. and  $V$  be an arbitrary input. A one-way function  $H(\cdot)$  satisfies the following:

1. Given  $x$ , it is easy to compute  $H(x)$ .
2. Given  $y$ , it is hard to compute  $x$  such that  $H(x) = y$ .

The first key that is used is  $k_n$ . When it expires, it is replaced with  $k_{n-1}$ . Likewise,  $k_j$  is replaced by  $k_{j-1}$  until  $k_0$  expires. At this point, there Alice and Bob run out of keys and a new  $k_0$  needs to be agreed upon. Thus  $n$  should be sufficiently large for the time scale of communications between Alice and Bob. The security of this scheme lies in the one-way property of  $H(\cdot)$  so that knowledge of the current key does not reveal future keys. However, prior keys are easily derived by the same property.

## 6.7 Conclusion

We have demonstrated and proved properties of key exchange based on random Markov models. The method is shown to generate highly random keys while remaining lightweight in terms of communication, storage, and computation costs.

We also note that the usefulness of the method is not restricted to keys, but to any variable which is periodically changed in a synchronous manner. For example, we have discussed the application to frequency-hopped communications systems. Of particular interest for this thesis is application to the multi-carrier authentication system in choosing the tag locations.

## Chapter 7

### Future Directions

We envision three main directions in which to take the current research.

First, with the upcoming adoption and availability of LTE and WiMax networks, it will be interesting to study how the physical layer authentication methods presented in chapters 3 and 4 may be incorporated. Specifically, there are questions of how the method work with existing authentication methods at the higher layers such as EAP (extensible authentication protocol): how do the higher layers interpret the authentication decisions generated at the physical layer?

Second, we can study how our method may be used together with other physical layer authentication methods. For example, the SEVILLE project [50] uses the fact that the wireless channel decorrelates rapidly with location to distinguish between authentic and inauthentic nodes. This is in contrast with the method presented in this thesis, where the transmission signals are perturbed to signal authentication information. The combination of the two methods may prove to be even more useful and powerful.

Third, rather than intentionally perturbing the signal, we may study how unperturbed signals may already have unique characteristics that may permit identification. That is, the authentication information may be unintentional and may arise from unique device characteristics. To analyze this, it may be possible

to parametrize such unintentional perturbations in order to create a suitable signal model. Having such a model will allow us to capture and exploit the signal characteristics.

## BIBLIOGRAPHY

- [1] U. M. Maurer, “Authentication Theory and Hypothesis Testing,” *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1350–1356, Jul. 2000.
- [2] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 2001.
- [3] P. W. Shor, “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer,” *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, Nov. 1997.
- [4] A. L. Shimpi. (2004, Jul.) Intel’s 90nm Pentium M 755: Dothan investigated. [Online]. Available: <http://www.anandtech.com/cpuchipsets/showdoc.aspx?i=2129>
- [5] F. Chabaud and A. Joux, “Differential Collisions in SHA-0,” in *Proceedings of the CRYPTO ’98*, Aug. 1998, pp. 56–71.
- [6] X. Wang, X. Lai, D. Feng, H. Chen, and X. Yu, “Cryptanalysis for Hash Functions MD4 and RIPEMD,” in *Proceedings of the EUROCRYPT 2005*, May 2005, pp. 1–18.
- [7] X. Wang and H. Yu, “How to break MD5 and Other Hash Functions,” in *Proceedings of the EUROCRYPT 2005*, May 2005, pp. 19–35.

- [8] B. Schneier. (2008, Aug.) Adi Shamir's Cube Attacks. [Online]. Available: [http://www.schneier.com/blog/archives/2008/08/ad\\_shamirs\\_cub.html](http://www.schneier.com/blog/archives/2008/08/ad_shamirs_cub.html)
- [9] S. M. Bellovin and M. Merritt, "Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks," in *Proceedings of the IEEE Symposium on Security and Privacy*, May 1992, pp. 72–84.
- [10] H. V. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer, 1994.
- [11] K. Ramchandran, M. Vetterli, and C. Herley, "Wavelets, Subband Coding, and Best Bases," *Proc. IEEE*, vol. 84, no. 4, pp. 541–560, Apr. 1996.
- [12] H. S. Wang and N. Moayeri, "Finite-state Markov channel - A useful model for radio communication channels," *IEEE Trans. Veh. Technol.*, vol. 44, no. 1, pp. 163–171, Feb. 1995.
- [13] A.-J. van der Veen and L. Tong, "Packet Separation in Wireless Ad-Hoc Networks by Known Modulus Algorithms," in *Proceedings of ICASSP 2002*, May 2002, pp. III-2149–III-2152.
- [14] L. Tong, B. M. Sadler, and M. Dong, "Pilot-assisted wireless transmissions: general model, design criteria, and signal processing," *IEEE Signal Process. Mag.*, vol. 21, no. 6, pp. 12–25, Nov. 2004.
- [15] B. Hassibi and B. Hochwald, "How much training is needed in multiple-antenna wireless links?" *IEEE Trans. Inf. Theory*, vol. 49, no. 4, pp. 951–963, Apr. 2003.
- [16] K. L. Blackard, T. S. Rappaport, and C. W. Bostian, "Measurements and models of radio frequency impulsive noise for indoor wireless communications," *IEEE Trans. Signal Process.*, vol. 11, no. 7, pp. 991–1001, Sep. 1993.

- [17] P. Torio and M. G. Sanchez, "Novel Procedure to Determine Statistical Functions of Impulsive Noise," *IEEE Trans. Electromagn. Compat.*, vol. 47, no. 3, pp. 559–568, Aug. 2005.
- [18] M. Ghosh, "Analysis of the effect of impulse noise on multicarrier and single carrier QAM systems," *IEEE Trans. Commun.*, vol. 44, no. 2, pp. 145–147, Feb. 1996.
- [19] D. Middleton, "Non-Gaussian noise models in signal processing for telecommunications: New methods and results for Class A and Class B noise models," *IEEE Trans. Inf. Theory*, vol. 45, no. 4, pp. 1129–1149, May 1999.
- [20] E. L. Lehmann, *Testing Statistical Hypotheses*. New York: Springer, 1997.
- [21] R. E. Blahut, *Principles and Practice of Information Theory*. Reading, MA: Addison-Wesley, 1987.
- [22] M. Bellare and T. Kohno, "Hash function Balance and Its Impact on Birthday Attacks," in *Advances in Cryptology - EUROCRYPT '04*, 2004, pp. 401–418.
- [23] A. D. Wyner, "The wire-tap channel," *Bell System Technical Journal*, vol. 54, no. 8, pp. 1355–1387, 1975.
- [24] I. Csiszar and J. Korner, "Broadcast Channels with Confidential Messages," *IEEE Trans. Inf. Theory*, vol. 24, no. 3, pp. 339–348, May 1978.
- [25] U. M. Maurer and S. Wolf, "Secret-Key Agreement Over Unauthenticated Public Channels - Part III: Privacy Amplification," *IEEE Trans. Inf. Theory*, vol. 49, no. 4, pp. 839–851, Apr. 2003.

- [26] P. Yu, J. S. Baras, and B. M. Sadler, "Physical Layer Authentication," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 1, pp. 38–51, Mar. 2008.
- [27] P. K. Vitthaladevuni and M.-S. Alouini, "A recursive algorithm for the exact BER computation of generalized hierarchical QAM constellations," *IEEE Trans. Inf. Theory*, vol. 49, no. 1, pp. 297–3073, Jan. 2003.
- [28] E. Biglieri, J. Proakis, and S. Shamai, "Fading Channels: Information-Theoretic and Communications Aspects," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2619–2692, Oct. 1998.
- [29] M. Medard, "The effect upon channel capacity in wireless communications of perfect and imperfect knowledge of the channel," *IEEE Trans. Inf. Theory*, vol. 46, no. 3, pp. 933–946, May 2000.
- [30] M. Dong, L. Tong, and B. M. Sadler, "Optimal insertion of pilot symbols for transmissions over time-varying flat fading channels," *IEEE J. Sel. Areas Commun.*, vol. 52, no. 5, pp. 1403–1418, May 2004.
- [31] P. Yu, J. S. Baras, and B. M. Sadler, "Multi-Carrier Authentication at the Physical Layer," in *IEEE Workshop on Security, Privacy and Authentication in Wireless Networks*, Newport Beach, CA, Jun. 2008, pp. 1–6.
- [32] —, "On Optimal Power Allocation in Multicarrier Authentication Systems," in *Army Science Conference*, Orlando, FL, Jun. 2008, p. to appear.
- [33] J. L. Massey, "Shift-Register Synthesis and BCH Decoding," *IEEE Trans. Inf. Theory*, vol. 15, no. 1, pp. 122–127, Jan. 1969.
- [34] J.-J. van de Beek, M. Sandell, and P. O. Borjesson, "ML Estimation of Time and Frequency Offset in OFDM systems," *IEEE Trans. Signal Process.*, vol. 45, no. 7, pp. 1800–1805, Jul. 1997.

- [35] T. M. Schmidl and D. C. Cox, “Robust Frequency and Timing Synchronization for OFDM,” *IEEE Trans. Commun.*, vol. 45, no. 12, pp. 1613–1621, Dec. 1997.
- [36] J.-J. van de Beek, O. Edfors, M. Sandell, S. K. Wilson, and P. O. Borjesson, “On channel estimation in OFDM systems,” in *Proceedings of the IEEE Vehicular Technology Conference*, vol. 2, Chicago, IL, USA, Jul. 1995, pp. 815–819.
- [37] O. Edfors, M. Sandell, J.-J. van de Beek, S. K. Wilson, and P. O. Borjesson, “OFDM Channel Estimation by Singular Value Decomposition,” *IEEE Trans. Commun.*, vol. 46, no. 7, pp. 931–939, Jul. 1998.
- [38] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, 1991.
- [39] T. M. Cover, “Broadcast Channels,” *IEEE Trans. Inf. Theory*, vol. 18, no. 1, pp. 2–14, Jan. 1972.
- [40] R. D. Silverman, “A Cost-Based Security Analysis of Symmetric and Asymmetric Key Lengths,” RSA Laboratories, Bulletin 13, Nov. 2001.
- [41] R. L. Rivest, A. Shamir, and L. Adleman, “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems,” *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.
- [42] V. S. Miller, “Use of Elliptic Curves in Cryptography,” *Lecture notes in Computer Sciences (CRYPTO 85)*, vol. 218, pp. 417–426, 1985.
- [43] N. Koblitz, “Elliptic Curve Cryptosystems,” *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, Jan. 1987.

- [44] R. Tarjan, “Depth-first search and linear graph algorithms,” *SIAM Journal on Computing*, vol. 1, no. 2, pp. 146–160, 1972.
- [45] J. Cheriyan and K. Mehlhorn, “Algorithms for dense graphs and networks on the random access computer,” *Algorithmica*, vol. 15, no. 6, pp. 521–549, Jun. 1996.
- [46] C. Cooper and A. Frieze, “The size of the largest strongly connected component of a random digraph with a given degree sequence,” *Combinatorics, Probability and Computing*, vol. 13, no. 3, pp. 319–337, May 2004.
- [47] L. R. Welch, “Hidden Markov Models and the Baum-Welch Algorithm,” *IEEE Information Theory Society Newsletter*, vol. 53, no. 4, pp. 1,10–13, Dec. 2003.
- [48] W. Diffie and M. E. Hellman, “New directions in cryptography,” *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.
- [49] L. Lamport, “Password Authentication with Insecure Communication,” *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.
- [50]