PHYSICAL-LAYER SECURITY: PRACTICAL ASPECTS OF CHANNEL CODING AND CRYPTOGRAPHY

A Dissertation
Presented to
The Academic Faculty

by

Willie K. Harrison

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in
Electrical and Computer Engineering

School of Electrical and Computer Engineering
Georgia Institute of Technology
August 2012

PHYSICAL-LAYER SECURITY: PRACTICAL ASPECTS OF CHANNEL CODING AND CRYPTOGRAPHY

Approved by:

Professor Steven W. McLaughlin,
Advisor
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Professor Faramarz Fekri
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Professor David V. Anderson
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Professor Jeff S. Shamma
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Professor Christopher J. Peikert
School of Computer Science
Georgia Institute of Technology

Date Approved: 1 June 2012

for my family and our future...

ACKNOWLEDGEMENTS

Let me start at the beginning and thank my parents. They provided a glorious childhood for me and my siblings in a small town in northern Utah. They are extremely diligent and humble, and they raised their family to appreciate traditional Christian values. No matter where my career leads, if I can only duplicate their success with their children, then I will have no regrets. Thank you Mom and Dad.

Next, I want to say thank you to my Ph.D. advisor, Dr. Steven McLaughlin. Steve hired me into his group at a time when most faculty probably wouldn't have been willing to take on new students. So Steve, thank you for letting me join the group, for guiding me in my research, for opening up the world to me, for teaching me to write, for letting me teach your class on occasion, and for involving me with Whisper Communications. I will always be grateful for the many diverse experiences that made up my Ph.D., and you were the driving force behind nearly all of them.

Also, a very sincere thank you to the members of my Ph.D. dissertation committee: Dr. Faramarz Fekri, Dr. David Anderson, Dr. Jeff Shamma, and Dr. Chris Peikert. I appreciate the time you spent serving on my committee and the guidance you provided one-on-one to me. In that same vein, I should thank the other faculty at Georgia Tech, especially Dr. Jim McClellan and Dr. Greg Durgin.

I had the fabulous opportunity during my Ph.D. to work with Dr. João Barros at the University of Porto. Working in his group was an absolute pleasure. Thank you João for helping me discover a meaningful problem and publish it within three months. Thank you also for picking me, my family, and our eight suitcases up from the airport, for helping us find a place to live, for taking us to the beach, for feeding us pancakes, and for entertaining us with your family symphony.

During the last two years of my Ph.D., I started working with some of the best people I know in a very cool start-up company called Whisper Communications. Thank you Steve, Jeff, Cenk, and Demijan. I learned a lot from each one of you, and will always call you my friends.

Also to the students and postdocs I spent time with in Porto: J.P., João A., Saurabh, Paulo, Hannez, Luisa, Mari, Fausto, Tiago, Ian, Joao R., Diogo, Rui, and Gerhard. Thanks for reading my bills for me and telling me how to pay them (Fausto), helping me find contact solution (also Fausto), taking me to Bola em Jogo for *futebol*, teaching me about Portugal, and, of course, solving research problems with me.

Although I appear to be the last of the McLaughlins for now, I still assert that our research group was very active and productive. Some of my best friends from my years at Georgia Tech are my labmates: Matthieu, Demijan, Arun, Jiaxi, and Hyoungsuk. Thanks guys. I'm sure we'll stay in touch. Also, thank you to my friends from church who also happened to be at Georgia Tech: John, Marc, Jake, Chris, Jared, Joseph, and James.

Now, to my siblings: Rachel, Jeremiah, Terrah, and Anna, and to your families, thank you all. All of you came out to Atlanta at least once while we were at Georgia Tech. That meant a lot to Krista and me. To Krista's family: thank you for not killing me for taking Krista and our boys so far away for so long, and thank you for the visits. Dave is the only one who has seen *every* apartment we've lived in. Nanette only missed one, also a Herculean effort.

Having saved the best for last, I finally want to thank, from the bottom of my heart, my biggest support group. The group leader is my insanely beautiful wife Krista, who spends her time taking care of the roughest, toughest, gang of hoodlums in Atlanta: Parker, Lincoln, Henry, and baby girl (due date: July 30, 2012). You guys are absolutely awesome! I love you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

A multilayer security solution for digital communication systems is provided by considering the joint effects of physical-layer security channel codes with application-layer cryptography. We address two problems: first, the cryptanalysis of error-prone ciphertext; second, the design of a practical physical-layer security coding scheme.

To our knowledge, the cryptographic attack model of the *noisy-ciphertext attack* is a novel concept. The more traditional assumption that the attacker has the ciphertext is generally assumed when performing cryptanalysis. However, with the ever-increasing amount of viable research in physical-layer security, it now becomes essential to perform the analysis when ciphertext is unreliable. We do so for the simple substitution cipher using an information-theoretic framework, and for stream ciphers by characterizing the success or failure of fast-correlation attacks when the ciphertext contains errors.

We then present a practical coding scheme that can be used in conjunction with cryptography to ensure positive error rates in an eavesdropper's observed ciphertext, while guaranteeing error-free communications for legitimate receivers. Our codes are called stopping set codes, and provide a blanket of security that covers nearly all possible system configurations and channel parameters. The codes require a public authenticated feedback channel.

The solutions to these two problems indicate the inherent strengthening of security that can be obtained by confusing an attacker about the ciphertext, and then give a practical method for providing the confusion. The aggregate result is a multilayer security solution for transmitting secret data that showcases security enhancements over standalone cryptography.

CHAPTER I

INTRODUCTION

Cryptography—literally, *secret writing*—has been used as a means of securing private information for roughly 3500 years, dating back to 1500 B.C. in ancient Mesopotamia where scribes encrypted secret recipes for pottery glazes using substitution ciphers [2]. Wartime encryption followed shortly thereafter, with the ancient Spartans being the first to employ military cryptography [2]. Their system was known as the *scytale*, pictured in Figure 1. The apparatus in the figure was used to construct a permutation on the letters in the message. First, a scroll or leather strip was wrapped around a stick of known proportions. Then, the message was written across the wrappings. The intended recipient of the message possessed a stick of the same proportions, and thus, could wrap the leather strip around his own stick and easily read the message. Perhaps one of the most famous military commanders to employ cryptography was Julius Caesar, who used a shift cipher to encrypt his personal and military messages [2].

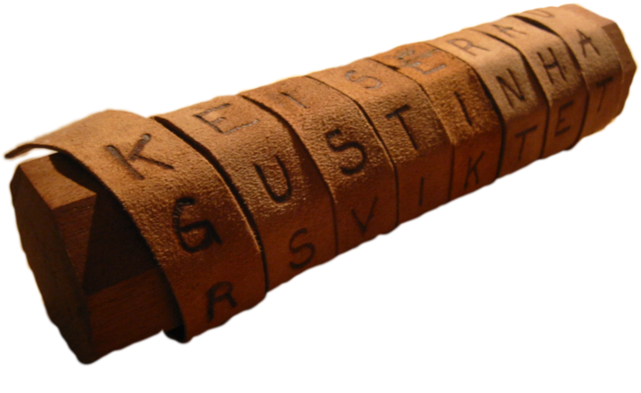

Figure 1: Depiction of a scytale, an instrument that provided ancient Spartans with a simple mechanism for performing permutation encryption and decryption. Figure obtained from [1].

The cryptanalysis of these early ciphers always assumes that an attacker has access to the complete encrypted message. After all, full knowledge of the cryptogram or ciphertext must have been available to an attacker because the very nature of ancient cryptanalysis required physically intercepting the sent message. Today, however, encrypted data are communicated over error-prone communication channels; therefore, this assumption no longer holds in general. In 1975, Aaron Wyner introduced the concept of the wiretap channel [3], and with it, the notion of physical-layer security. A modern version of the wiretap channel model is presented in Figure 2. The model depicts an entity named Alice who sends a secret message to a receiver named Bob through the *main channel* of communications, which is denoted Q_m. During data transfer, an eavesdropper named Eve overhears the transmitted message, although through a separate channel Q_w called the *wiretap channel*.

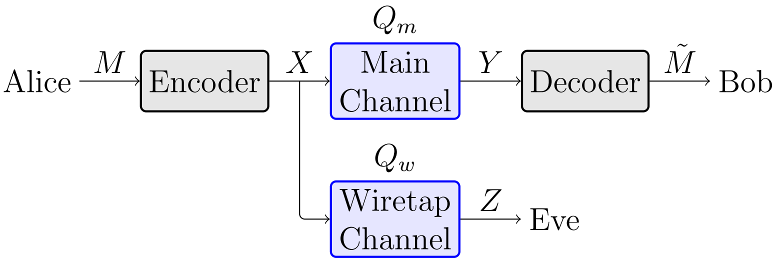

Figure 2: Wiretap channel model depicting Alice sending a message to Bob over the main channel Q_m, while Eve listens in on their communications over the wiretap channel Q_w.

Although traditional cryptanalysis models assume that the ciphertext is known to an attacker, the wiretap channel model can be used to consider cases where an eavesdropper only has access to error-prone data. Physical-layer security is then any security obtained by exploiting the physical characteristics, e.g., noise, of the communications channels. Since Wyner's paper in 1975, coding schemes have been found that take advantage of irregularities in communication channels for secrecy, although most of them are somewhat impractical for real world implementation. Our approach to security of transmitted data is considered a multilayer approach, as outlined in Figure 3. First, we wish to understand how the strength of a cryptosystem is affected when an attacker's ciphertext is contaminated with errors. Second, we wish to provide practical channel coding techniques that offer reliability for legitimate receivers, and exploit the noisy physical layer of a communications system to add confusion, or a positive error rate in the ciphertext, for eavesdroppers. Finally, we consider the joint security of coding with cryptography and discuss the benefits of relying on both types of encoding to keep data safe, rather than rely solely on one or the other.

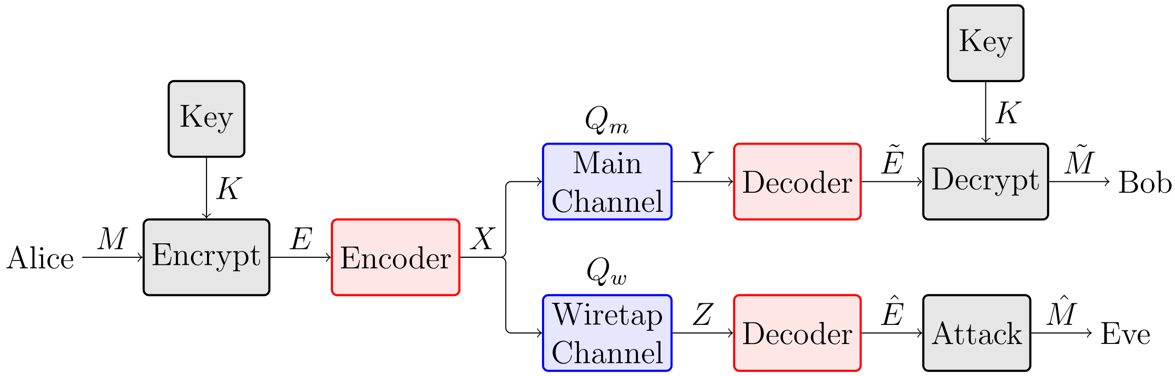

Figure 3: General overview of a multilayer approach to security including key-based encryption and decryption and keyless physical-layer encoding and decoding in a wiretap channel setting.

There is a danger in relying on only one layer of security, regardless of how strong that layer is considered to be. Typical arguments against cryptography as a standalone security solution are firstly, that modern ciphers are designed around the assumption that certain mathematical operations are hard to compute. For example, the factoring of large integers, is deemed a hard problem if the integers are large enough. However, no proof exists showing this problem to conclusively be difficult. We simply do not have efficient ways to solve this problem right now [4]. Furthermore, modern ciphers can often be attacked through side avenues when users employ the systems incorrectly, or using social engineering, thus relieving attackers from the necessity of attacking the assumed hard problems by inadvertently opening other ways for them to obtain the data. Finally, there is nothing to stop entities from accidentally making use of outdated ciphers with known efficient attacks.

Of course, it would be foolish to reject cryptography because of these potential weaknesses. Let us say, that instead of cryptography, we now wish to rely solely on physical-layer coding techniques for data security. After all, the security measures are information-theoretic, and information-theoretic security is now commonly accepted as the strictest form of security [4]. However, there are many systems in the real world that provably cannot benefit from physical-layer security. For example, in a system without feedback, if an eavesdropper has a better channel than a legitimate receiver, physical-layer security schemes will either offer no protection or possibly limited protection depending on how much information the transmitting party knows about the system. Clearly then, relying on physical-layer security exclusively, is not a viable option for security in many instances.

We suggest that the proper implementation of physical-layer security in modern systems provides an enhancement to already-encrypted data. Again, a multilayered approach to security can provide the benefits of both physical-layer security and cryptographic security in the same system. The two layers are extremely complimentary to one anther, as will be seen throughout this work. Furthermore, physical-layer security schemes are often keyless, relying only on channel characteristics for secrecy. Since secret key distribution can be somewhat painful in a practical system, it makes sense to look into physical-layer security techniques as a possible simplification in the key requirements of a system.

As a roadmap through the remaining chapters of this dissertation, we provide the following explanation. It will first be beneficial to provide a system-level overview as well as set forth some general notation, and then present some basic background in information theory, cryptography, channel coding, and physical-layer security. All of this is done in Chapter 2. In addressing the issue of cryptanalysis of imperfect ciphertext, we look at characterizing the security enhancement that can be obtained if channel coding can simply provide a positive error rate in the eavesdropper's received data stream. In Chapter 3 we analyze the simple substitution cipher in an information-theoretic sense when symbols of ciphertext are erased at random through a communications channel. The security enhancement from the erasures is given in terms of equivocation, or conditional entropy. Since substitution ciphers are often included in more modern and complex ciphers, this analysis may provide insights into a number of current cryptosystems. Then, thinking more practically, we investigate stream ciphers in Chapter 4, providing insight into how cryptosystems can be individually analyzed based on existing attacks to chart the effects of errors in the ciphertext.

Following our discussions on cryptanalysis of noisy ciphertext, we move on to a novel practical physical-layer channel coding scheme in Chapter 5 that exploits packet erasures in the wiretap channel for secrecy. The coding at the physical layer is based on low-density parity-check (LDPC) codes, and assumes authenticated public feedback channels for all legitimate receivers of the transmitted data. Several encoding and packaging techniques are leveraged for secrecy within the design to provide a *blanket* of security that covers almost all possible channel parameter configurations. We move on to further discuss the implications of combined security from cryptography and the physical-layer in Chapter 6. Initial findings of the interplay between channel coding for secrecy and cryptography are encouraging in that for codes and ciphers analyzed, we have been able to characterize physical-layer security by noting the reduced effectiveness of known attacks, or the increase in general confusion, given a certain percentage of errors at the eavesdropper's receiver. We also note that our proposed encoder and decoder provide an error rate in excess of the decoding threshold to an eavesdropper while maintaining reliable communication with legitimate parties using feedback. We also provide some conclusions and discuss future work that may stem from this research in Chapter 6.

As a guide through the publications of the author, Chapter 3 is drawn almost exclusively from [5]. Chapter 4 is primarily comprised of material from [6, 7] and [8]. The physical-layer coding scheme in Chapter 5 can be found in the literature in [9, 10] and [11]. Each of these papers, to some degree, express the idea of combined security addressed in Chapter 6.

CHAPTER II

PRELIMINARIES

As a precursor to addressing the problem of combined security derived from both cryptography and physical-layer security, it is first requisite to provide a system-level overview of the general setup, and set forth some notation regarding that system. In addressing the big picture up front, it is easier to see how each result contributes to the overall goal of this work, that is, to provide combined security through cryptography and physical-layer security coding. It is also necessary to give some basic background in information theory, cryptography, channel coding, and physical-layer security. When discussing physical-layer security in this chapter, we also discuss some of the short-comings of specific code designs that offer security, and introduce a new metric, degrees of freedom, that may be useful in assessing the security of some physical-layer schemes.

2.1 System-Level Overview and Notation

In a typical digital communications system, we anticipate a setup similar to that shown in Figure 4. We see a source of data, say Alice, and a destination for that data to be transmitted or sink, say Bob. Assume that Alice's data are discrete symbols from a generic alphabet $\mathcal{A}$. Prior to transmission of Alice's data, she may choose to pass the data through a series of encoders. Figure 4 depicts three encoders; namely, a source encoder, a cryptographic encoder, and a channel encoder [12]. Of course, Bob's receiver possesses matching decoders in reverse order. The source encoder removes redundancy in the data through some compression algorithm [13]; the cryptographic encoder conceals the meaning of the data from potential attackers of the system [14]; and finally, the channel encoder adds redundancy back into the data for the purpose of error detection and correction [12]. The research presented in this work deals primarily with the cryptographic encoder and the channel encoder. Thus, we often assume the data have been compressed, but will not offer details as to the compression techniques.

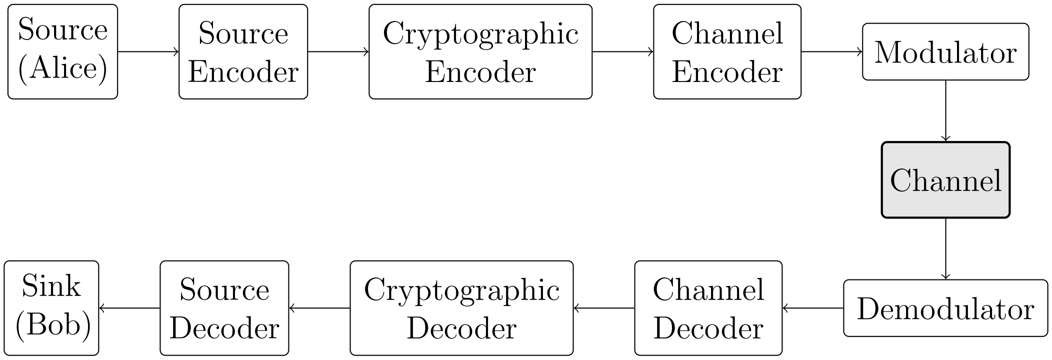

Figure 4: Typical digital communications system with a series of encoders and decoders.

Throughout this work, we will follow the signal conventions and basic general design portrayed in Figure 5. The figure shows a compressed message M as the input to a cryptographic encoder with cryptogram E as the output. The signal is then encoded and packetized to obtain a collection of packets X for transmission. Bob receives a collection of packets Y through the main channel Q_m, while Eve receives packets Z through the wiretap channel Q_w. Both channels are assumed to be packet erasure channels (PECs), meaning the receiver either receives full information about a packet or zero information about a packet with some fixed probability. The design of the encoder in Chapter 5 exploits the nature of packet erasure channels (and may also be applicable to other types of channels), and magnifies errors in the decoder caused by missing packets, all the while providing reliable communications to Bob using an authenticated feedback channel for automatic repeat request (ARQ). To be clear, all traffic on the feedback channel is public knowledge, so Eve can listen to everything occurring on the channel. However, since the channel is authenticated, Alice can detect whether or not transmissions on the channel come from a trusted source. This effectively reduces Eve to passive status. Erasures occur with probability δ in the main channel and with probability ϵ in the wiretap channel. Both Bob and Eve attempt to decode the data to obtain the cryptogram E. Bob's decoder output is denoted $\tilde{E}$, and Eve's is $\hat{E}$. Finally, Bob decrypts using the known secret key K and obtains $\tilde{M}$. Eve does not know the secret key, and thus, must attack the cryptogram to obtain $\hat{M}$, her best estimate of the message. The system design guarantees that the decoded and decrypted message $\tilde{M}$ for Bob is error free so that $\Pr(\tilde{M} \neq M) = 0$, while guaranteeing decoding failure for Eve with high probability for almost all channel state parameter pairs (δ, ϵ). Clearly Eve can conceivably obtain partial information about the message in such a context. The encoder design, however, seeks to mitigate the usefulness of any leaked information in an attack.

Given this setup, it will be shown in Chapters 5 and 6 how missing packets at the eavesdropper propagate incorrect bit assignments through the system so that the decoded ciphertext is extremely unreliable. As a result, normally successful attacks against the cryptographic layer fail reliably because of excessive error rates in the ciphertext. This problem of cryptanalysis with error-prone ciphertext is actually treated first in Chapters 3 and 4. Eve can be made to receive error-prone ciphertext, even when she has a better channel than Bob, because Bob can request missing packets. Since the feedback channel is authenticated, Eve cannot make requests of her own. The intersection between the set of dropped packets in Q_m and the set of dropped packets in Q_w may yet be obtained by the eavesdropper during retransmissions; however the packets dropped in Q_w that are not dropped in Q_m are forever lost to Eve.

It is also worth noting that since chapters in this dissertation address either noisy ciphertext cryptanalysis or the design of physical-layer security codes, we will find it advantageous to ignore portions of Figure 5 for chapters at a time. However, it should be noted, that the overall picture is still crucial to the interplay between the

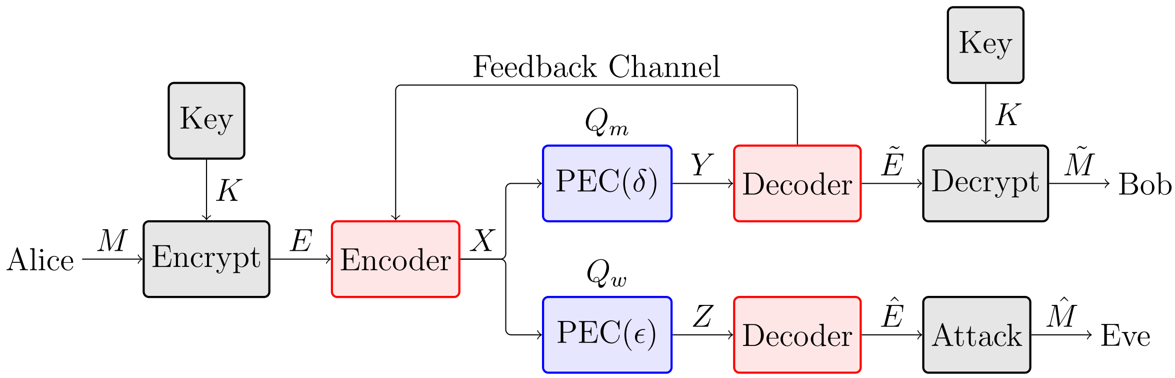

Figure 5: General overview of a multilayer approach to security including cryptography and physical-layer security, in a packet-loss environment with authenticated feedback for Bob.

different layers of security.

Regarding other basic notation, the manuscript will adhere to the following rules throughout the document.

x	A scalar (lowercase).
$\mathbf{x}^n$	A length-n row vector (bold lowercase, superscript may be omitted).
X	A random variable, set, event, etc. taken in context (uppercase).
$\mathbf{X}^n$	A length-n random vector (bold uppercase, superscript may be omitted).
$\mathbf{X}^{m\times n}$	An $m \times n$ matrix (bold uppercase, superscripts may be omitted).
$\mathcal{X}$	An alphabet of discrete symbols (caligraphy).
$\Pr\left(X\right)$	The probability of event X.
$\mathbb{E}[X]$	The expectation of random variable X.
$p_X(x)$	The probability mass function (pmf) of discrete random variable X.
$\mathbb{H}(X)$	Shannon entropy of discrete random variable X (in bits). See Definition 1.
$\mathbb{I}(X;Y)$	The mutual information between discrete random variables X and Y. See Definition 3.

2.2 Information Theory

Physical-layer security is a research area in information theory, and hence, most of the traditional security metrics are information-theoretic. Hence, we provide here a brief overview of basic information-theoretic quantities, and refer the interested reader to [13] for more details. For the following definitions, assume X and Y to be discrete random variables with respective pmfs $p_X(x)$ defined over $\mathcal{X}$ and $p_Y(y)$ defined over $\mathcal{Y}$.

Definition 1. The *entropy* of X is given as

$$\mathbb{H}(X) = -\sum_{x \in \mathcal{X}} p_X(x) \log_2 p_X(x).$$

The entropy can be thought of as a measure of the uncertainty in guessing realizations of a random variable. Therefore, if a value in $\mathcal{X}$ occurs with probability zero, then the entropy that value contributes to the total entropy of X is also zero. This makes sense, because an event or value that never occurs cannot possibly increase our uncertainty of the random variable. Specifically, all information-theoretic definitions assume $0 \cdot \log 0 = 0$. Also note that all of the logarithms in this work are base two, and thus, all information-theoretic quantities are measured in *bits*.

Definition 2. The conditional entropy of X given Y is defined as

$$\mathbb{H}(X|Y) = \sum_{y \in \mathcal{Y}} p_Y(y) \mathbb{H}(X|Y = y).$$

Intuitively, $\mathbb{H}(X|Y)$ is a measure of the uncertainty in X that is not shared by Y, or the uncertainty that remains in X if Y is known.

Definition 3. The *mutual information* between X and Y is calculated as

$$\mathbb{I}(X;Y) = \mathbb{H}(X) - \mathbb{H}(X|Y).$$

This quantity is equal to the amount of information that X and Y share, or the amount of information that one can theoretically collect about one if you know the other. Mutual information is symmetric in that $\mathbb{I}(X;Y) = \mathbb{I}(Y;X)$.

Definition 4. A *discrete memoryless channel* (DMC) is a channel with input modeled by the discrete random variable X and output modeled by the discrete random variable Y where the probabilities of specific outputs in Y are determined by transition probabilities $p_{Y|X}(y|x)$. The memoryless aspect of the channel indicates that the ith channel output is only a function of the ith channel input.

Definition 5. For a communications channel, if X is the input and Y is the output, then the *channel capacity* C is defined as

$$C = \max_{p_X(x)} \mathbb{I}(X;Y). \tag{1}$$

Note that X and Y are still deemed to be discrete random variables, therefore, the channel with capacity C can be described by transition probabilities $p_{Y|X}(y|x)$. The intuitive notion of channel capacity is the highest encoding rate that the channel can support with vanishingly low probability of error. This is discussed further in Section 2.4. The maximization is over all possible distributions on X. Therefore, there may be some inputs that cannot achieve the channel capacity due to their distributions.

2.3 Cryptography and Perfect Secrecy

Many cryptosystems in place today measure security computationally. If all attacks are computationally intractable, then the system is deemed to be secure. The chief failings of this notion of security are the assumptions placed on the attacker. First, it is assumed that the attacker has limited resources to confront the problem, even if those resources are state of the art. Second, new and unanticipated algorithmic attacks can be developed against the conjectured hard problems. Claude Shannon addressed these shortcomings by defining the notion of perfect secrecy [15]. This was the first attempt at measuring security with an information-theoretic metric. Information-theoretic security makes no computational assumptions on the attacker, and is accepted as the strictest form of security [16].

Definition 6. If a secret message M is encrypted to form a cryptogram E using a secret key K, then *perfect secrecy* is achieved if

$$\mathbb{H}(M|E) = \mathbb{H}(M), \tag{2}$$

that is, if the ciphertext provides no information about the message. Note that the entropies in (2) are calculated assuming K is chosen according to some random key distribution (usually uniform).

We note here, that the quantity $\mathbb{H}(M|E)$ is typically called the *message equivocation*, and $\mathbb{H}(K|E)$ is called the *key equivocation*.

2.3.1 One-Time Pad

Shannon analyzed the one-time pad cipher shown in Figure 6, and found that it achieves perfect secrecy. This cipher uses a key K composed of uniformly random binary data, and encrypts M by the operation

$$E = M \oplus K, \tag{3}$$

where $\oplus$ signifies a bitwise exclusive-or (X-OR) function. The decoding assumes a noiseless channel and combines the same key with the received data $\hat{E}$ to recover the message

$$\hat{M} = \hat{E} \oplus K. \tag{4}$$

This type of cryptography is *symmetric*, as the same key and operation perform both encryption and decryption. Although the one-time pad attains perfect secrecy, it fails to solve the practical problem of distributing the message, because $|K| = |M|$, and

K must still be distributed secretly. In fact, Shannon proved that perfect secrecy is only attainable if the key is at least as long as M, or more generally, only if

$$\mathbb{H}(K) \geq \mathbb{H}(M). \tag{5}$$

Although Shannon's result seems disheartening, perfect secrecy also makes the limiting assumption that an attacker has access to an error-free cryptogram; however, this may not be the case in practice.

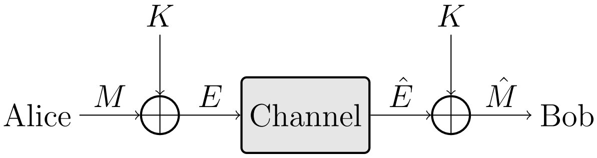

Figure 6: One-time pad.

2.4 Channel Coding

Channel codes are typically designed to make communications more reliable by adding redundancy into transmitted data that allow for error detection and correction at the receiver. Channel coding is also typically the last encoding rule prior to transmission as shown in Figure 4, thus preventing the propagation of errors at the decoder, because errors are corrected in the first step at the receiver. Since redundancy is added to the data during channel encoding, the length of the data stream increases according to the rate of the code.

Definition 7. Let k be the number of input bits to the encoder, and n be the number of output bits from the encoder. Then the *rate* of the code is

$$R = \frac{k}{n}, \tag{6}$$

and the code is called an (n, k) code.

Thanks to the father of information theory, Claude Shannon, we know that for rates less than the channel capacity C there exist channel codes that can obtain arbitrarily low probability of decoding error [17].

2.5 Physical-Layer Security

For physical-layer security code design, we not only wish to obtain arbitrarily low probability of decoding error for Bob, as in traditional channel coding, but also wish to provide some level of security against Eve.

2.5.1 Theory of Physical-Layer Security

The theoretical basis for physical-layer security is derived from the wiretap channel model, and then includes a number of different information-theoretical metrics for security. The secrecy capacity of a system is then defined based on satsfying those two goals. Our research also addresses regions of information leakage that do not necessarily fit within established security metrics. Therefore, we also introduce a new metric that allows us to account for information-theoretic security and cryptographic security in tandem.

2.5.1.1 Wiretap Channel Model

In 1975, Aaron Wyner gave birth to physical-layer security with his *degraded* wiretap channel model shown in Figure 7, along with a new condition for secrecy [3]. Let a message M of length k be encoded into a codeword X of length n, and then transmitted. A legitimate receiver obtains Y over the *main channel* denoted Q_m, and an eavesdropper obtains Z, a degraded version of Y, through an additional channel called the *wiretap channel* Q_w.

Definition 8. The secrecy condition is

$$\lim_{n \to \infty} \frac{\mathbb{I}(M;Z)}{n} = 0, \tag{7}$$

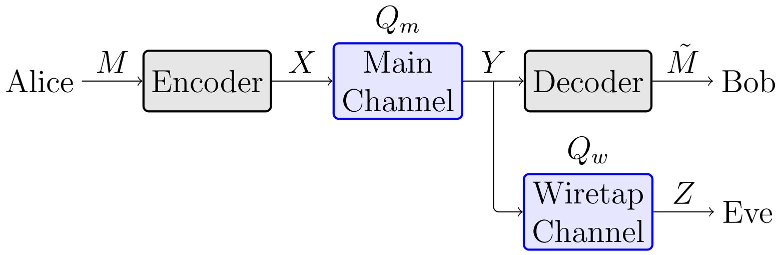

Figure 7: Degraded wiretap channel model depicting Alice sending a message to Bob over the main channel Q_m, while Eve observes Bob's received data through yet another channel Q_w.

and is termed today *weak secrecy*.

Weak secrecy requires the rate of information leaked to an eavesdropper to go to zero as the blocklength of the encoder gets large. Wyner showed that for rates up to the *secrecy capacity* C_s, encoders and decoders exist that satisfy (7) and also achieve arbitrarily low probability of error for intended parties in this degraded case, i.e. when $X \to Y \to Z$ is a Markov chain. Csiszár and Körner [18] later generalized these results removing the degraded restriction, but still showed that $C_s > 0$, only if Q_m is *less noisy* than Q_w. This more general version of the wiretap channel model was already shown in Figure 2 in Chapter 1. For the general degraded wiretap channel, the secrecy capacity is [4]

$$\begin{aligned} C_s &= \max_{p_X(x)} \mathbb{I}(X;Y|Z) && (8)\\ &= \max_{p_X(x)} (\mathbb{I}(X;Y) - \mathbb{I}(X;Z)) && (9)\\ &\geq \max_{p_X(x)} (\mathbb{I}(X;Y)) - \max_{p_X(x)} (\mathbb{I}(X;Z)) && (10)\\ &= C_m - C_w, && (11)\end{aligned}$$

where C_m is the channel capacity of the main channel, and C_w is the channel capacity of the wiretap channel. Note, for some channels, $C_s = C_m - C_w$, but in the general degraded wiretap channel, the secrecy capacity is at least equal to this difference. Intuitively then, the greater the advantage in channel quality that a legitimate receiver can leverage over an eavesdropper, the higher the rate at which data can be encoded but remain secret using only physical-layer security.

It is of some relevance to point out that there is a more meaningful secrecy condition besides that of weak secrecy.

Definition 9. If a system is such that

$$\lim_{n\to\infty} \mathbb{I}(M;Z) = 0, \tag{12}$$

then the system achieve strong secrecy.

Clearly this condition is much more restrictive on the information leakage than is the weak secrecy condition in (7). Also, one may expect C_s to decrease if we define secrecy capacity as the maximum encoding rate at which strong secrecy is achievable, rather than weak secrecy. However, it is a well-known fact that for the degraded wiretap channel model, the secrecy capacity is the same regardless of which condition is used [19]. For more details as to the theory of physical-layer security, we direct the interested reader to the first concise text on the matter [4]. It should also be noted that semantic security addresses practical issues beyond those regarded in these information-theoretic security definitions [20].

2.5.2 Practice of Physical-Layer Security

Significant advances in the understanding of theoretically achievable secrecy rates of communication systems have been made beyond those already outlined—again, see [4] for further details. However, another of the main challenges regarding physical-layer security has been the design of practical schemes that achieve the secrecy rates indicated by the theory. These schemes exploit noise in the channel at the physical layer of the communications system to derive and magnify an advantage over the eavesdropper.

2.5.2.1 Notable Schemes and Techniques

The historical development of practical secrecy coding schemes began with Wyner in his original wiretap paper. His design was also clarified in an extension of the original work [21]. Here the general idea of partitioning a group code into cosets to achieve secrecy was first presented. Wei noted in 1991 [22] that generalized Hamming weights (which are defined based on the minimum-sized *support* for subcodes of linear codes) can be used to characterize the performance of a linear code for the channel model depicted in [21]. The coset coding technique was shown to apply to low-density parity-check (LDPC) codes (see [23] and [24]) much more recently in [25], where code designs are given that achieve the weak secrecy condition in (7) for noiseless main channels when the wiretap channel is a binary erasure channel (BEC). This work in LDPC codes for secrecy has been furthered in [26, 27], where large-girth LDPC codes are considered, and shown to meet the strong secrecy constraint in (12) in certain cases. These codes always satisfy the weak secrecy condition when Q_m is noiseless, and Q_w is a BEC. Some interesting recent work discusses polynomial-time algorithms that provide semantic security to some systems [28].

Interleaving coded symbols has been used in [29] and [30] in conjunction with wiretap codes developed in [25] to offer secrecy to various systems. The secrecy properties of nonsystematic LDPC codes have also been discussed in [31]. The works of [32, 33] and [34] show how LDPC codes can be punctured to increase secrecy for scenarios where legitimate parties have an advantage in channel quality over eavesdroppers. Finally, it should be noted that Arıkan's polar codes [35] can offer secrecy for general symmetric channels, although code construction is an issue for non-erasure channels. Schemes have been presented in [36] and [37] that achieve weak secrecy, although these schemes only offer secrecy for degraded wiretap channels. Furthermore, design of these codes is heavily contingent on perfect channel state information (CSI) at the encoder. There remains a need to deliver schemes that are robust enough to offer security, even when system parameters are unknown to the designer.

2.5.2.2 Limitations of Current Schemes

While practical codes exist that obtain varying levels of information-theoretic security, most designs suffer from one or more of several drawbacks. For instance, code designs are almost always a function of specific channel parameters, or CSI, seen by legitimate receivers and eavesdroppers. Therefore, channels with varying or unknown parameters present design issues. For example, in the case of an undetected eavesdropper, a designer would have no information as to the channel parameters in Q_w. Furthermore, say an eavesdropper listens to wireless communications from a different location over time. Then even if the eavesdropper could be detected, we would need time varying codes to take advantage of this information.

Other codes offer secrecy for only specific types of channels, or only when the eavesdropper's channel is degraded. Still other designs are impractical in the real world as a result of design complexity, necessary side information for legitimate decoding, or other limitations. Finally, the most glaring shortcoming of any scheme that derives security from the physical layer of a communications system, is that if an eavesdropper has a *better* channel than a legitimate receiver, say the channel capacity of the wiretap channel C_w exceeds that of the main channel C_m, then the scheme will likely fail. For some channels, $C_w > C_m$ implies that the secrecy capacity of the system is zero. For example, consider what occurs when an eavesdropper has a noise-free channel and $Z = X$. These types of scenarios necessitate the coupling of physical-layer security schemes with additional protection. Of course, cryptography can fill that role nicely.

2.5.2.3 Degrees of Freedom

For this reason, and as part of a preview of coming attractions, we want to consider cases where the goal of the encoder is not necessarily to provide weak, strong, or perfect secrecy, but rather a constant positive error rate in an eavesdropper's received data. Assuming this data stream to be error-prone ciphertext provides a reasonable model for comprehending the effects of physical-layer security coupled with cryptography. It should be noted that this notion of security is in fact weaker than strong secrecy and weak secrecy, but may provide a more practical context from which to view the multilayer security problem.

Suppose that it is possible to encode already-encrypted data such that a legitimate party receives the cryptogram with no errors, but an eavesdropper receives the cryptogram with no information about some symbols. If there is truly no information about these symbols available to an eavesdropper, they will be forced to attempt a brute-force attack on this subset of symbols, that is, attacking the system by simply guessing unknown values. Let us call these symbols for which an eavesdropper recovers no information *degrees of freedom* D, because they represent degrees of freedom in the cryptogram space for the eavesdropper [10].

As a comparison between D and more conventional security notions, if bits in E are uniformly zero or one and independent and identically distributed (i.i.d.), then perfect secrecy implies $D = k$, where k is the dimension of the encoder. This means that there are exactly 2^k equally likely binary length-k sequences with which to fill in the missing data. Note that an encoder only has 2^k possible codewords in the code. Thus, every codeword is equally likely *a posteriori* to the eavesdropper in a perfectly secure setup. In fact, on average there are $\mathbb{E}[2^D]$ equally likely binary codewords in a maximum a posteriori (MAP) decoder. This implies a multiplication of efforts necessary to attack a cryptogram with D degrees of freedom.

If the degrees of freedom are the only source of confusion to Eve, then the equivocation (measured in symbols) of the transmitted message X and the eavesdropper's received data Z is related to D in that $\mathbb{H}(X|Z) = \mathbb{E}[D]$. Therefore, we see that degrees of freedom can provide the answers to meaningful practical security questions, as in how much stronger a system is if physical-layer security is employed to enhance cryptography, but can also map back to information theoretic origins.

CHAPTER III

SIMPLE SUBSTITUTION CIPHER WITH ERASURES

Cryptanalysis is the process of computing the secret key or message of a cryptosystem given the ciphertext [14]. Our problem requires estimating cryptographic strength when the ciphertext has errors. Since security analysis of cryptographic systems is aimed at proving the robustness of the system, a worst-case assumption that the attacker has perfect ciphertext is traditionally made. Therefore, the analysis of the strength of cryptography assuming errors in the ciphertext is a virtually untouched research topic. Only now, under the framework of physical-layer security, does it become a viable topic for investigation. In this chapter, we investigate the simple substitution cipher when ciphertext symbols are erased randomly at the eavesdropper's receiver. This chapter is derived from [5] and has been submitted for publication.

3.1 Motivation

In Shannon's landmark paper on the communication theory of secrecy systems [15], he discussed the secrecy of a simple substitution cipher. The encryption outputs a cryptogram (ciphertext) E as a function of a message (plaintext) M and a key K. His analysis provided the key equivocation $\mathbb{H}(K|E)$ for the simple substitution cipher in general terms, and showed that $\mathbb{H}(K|E)$ and the message equivocation $\mathbb{H}(M|E)$ are appropriate metrics for the characterization of the strength of ciphers. Later, Blom [38] and Dunham [39] completed the specific calculations of $\mathbb{H}(K|E)$ and $\mathbb{H}(M|E)$, respectively, for the simple substitution cipher, while additional security characterizations were made by Sgarro in [40]. A common assumption made in all of these initial works is that an attacker always has access to a clean version of E.

However, due to the contributions of Wyner [3] and others [18] as outlined in

Chapter 2, we know that *security* against eavesdroppers can be formulated another way that takes into account characteristics of a noisy communications channel through which an eavesdropper obtains transmitted data. Many coding schemes were mentioned in Section 2.5.2.1 that are commonly called *wiretap codes.* These and others like them, were developed to offer security by exploiting physical-layer characteristics. In Chapter 5, we showcase our own code designs that were developed to further secure already-encrypted data by structuring the leaked information such that attacks are restricted to a brute-force search over a subset of symbols for which the attacker has no information. In other words, the code gives degrees of freedom in Eve's cryptogram as discussed in Section 2.5.2.3.

In a real system, sufficient signal reliability at the eavesdropper may, however, undermine physical-layer attempts at secrecy coding. Thus, we continue to stress that wiretap codes should be coupled with cryptographic schemes for a more complete security solution. This coupling removes the commonly assumed requirement that codes must individually provide secrecy, but rather allows coding to act as a security enhancement to cryptography. Suppose codes simply confuse attackers by providing a positive error rate in an attacker's eavesdropped cryptogram. The security implications of such a code can then be characterized through cryptanalysis of noisy ciphertext.

Chapter 4 will show this analysis for correlation attacks on stream ciphers. This chapter will provide the analysis for the simple substitution cipher. The cryptanalysis performed herein is information-theoretic, and addresses the setup of an eavesdropper obtaining ciphertext through an erasure channel. The results are compared to the analysis when the attacker always has an error-free version of the cryptogram [38, 39]. The comparison quantifies the increase in security obtainable by wiretap coding. Since substitution ciphers are often a significant piece in more modern cryptosystems, this analysis may also carry over to more current ciphers. Furthermore, this analysis can be seen as a first look at the noisy-ciphertext attack for block ciphers.

Regarding the rest of the chapter, notation specific to Chapter 3 is set forth in Section 3.2, while Section 3.3 provides the background to the substitution cipher and the packet erasure channel model. Section 3.4 then calculates the key equivocation and the message equivocation for the cipher assuming erased symbols, and the results are discussed by way of conclusion in Section 3.5.

3.2 Notation

Recall $\mathcal{X} = \{1, 2, \ldots, N\}$ denotes a finite set, or alphabet of discrete symbols. A length-L vector of symbols from $\mathcal{X}$ is denoted $\mathbf{x}^L = (x_1, x_2, \ldots, x_L) \in \mathcal{X}^L$. Throughout the chapter, however, it is often more useful to consider symbols as they occur randomly, and hence we write

$$\mathbf{X}^L = (X_1, X_2, \ldots, X_L), \tag{13}$$

to signify a random vector of message symbols drawn from $\mathcal{X}$ according to a probability mass function (p.m.f.) $p_X(x)$. The superscript on vectors may be omitted when L is obvious.

Furthermore, consider that vectors $\mathbf{x}^L$ and $\mathbf{y}^L$ are composed of symbols from some alphabet, and then elements from one or more of the vectors are randomly erased, i.e., no information is known about them. Let $\hat{\mathbf{y}}^L$ and $\hat{\mathbf{x}}^L$ be the resulting output vectors. If the ith symbol in $\hat{\mathbf{x}}^L$ is erased, we write $\hat{x}_i = e$. Let the notation

$$\hat{\mathbf{x}}^L \overset{e}{=} \hat{\mathbf{y}}^L \tag{14}$$

signify that $\hat{x}_i = \hat{y}_i$ for every i where both $\hat{x}_i \neq e$ and $\hat{y}_i \neq e$. Thus, if (14) holds, the equality of $\mathbf{x}^L$ and $\mathbf{y}^L$ can only be determined if both $\hat{\mathbf{x}}^L$ and $\hat{\mathbf{y}}^L$ have zero erasures.

At times, we will find it easier to work with patterns of symbols, rather than the symbols themselves. Borrowing notation from [41], we define the *index* $\imath_{\mathbf{x}^L}(x)$ to be one more than the number of distinct symbols occurring in $\mathbf{x}^L$ prior to the first appearance of x. The *pattern* of $\mathbf{x}^L$ is then given as the concatentation

$$\Psi(\mathbf{x}^L) = \imath_{\mathbf{x}^L}(x_1)\imath_{\mathbf{x}^L}(x_2)\ldots\imath_{\mathbf{x}^L}(x_L). \tag{15}$$

For example, if $\mathbf{x} = (2, 3, 5, 2, 4, 2, 5, 2, 2, 1)$, then $\Psi(\mathbf{x}) = 1231413115$ because $\imath_{\mathbf{x}}(2) = 1$, $\imath_{\mathbf{x}}(3) = 2$, $\imath_{\mathbf{x}}(5) = 3$, $\imath_{\mathbf{x}}(4) = 4$, and $\imath_{\mathbf{x}}(1) = 5$.

Since this chapter deals only with the substitution cipher, all length-L vectors that have the same pattern are said to belong to the same *residue class*. This term is properly defined in Section 3.3.1 with more background on the substitution cipher. For now, suppose these residue classes are indexed, and again take the vectors with possible erasures $\hat{\mathbf{x}}^L$ and $\hat{\mathbf{y}}^L$. We write

$$\Psi(\hat{\mathbf{x}}^L) \overset{e}{=} \Psi(\hat{\mathbf{y}}^L). \tag{16}$$

to indicate that the erasures in $\hat{\mathbf{x}}^L$ and $\hat{\mathbf{y}}^L$ prevent us from knowing whether or not $\Psi(\mathbf{x}^L) = \Psi(\mathbf{y}^L)$.

Finally, let us also denote $S(\mathbf{x}^L)$ as the number of unique symbols in $\mathbf{x}^L$. For example, if $\mathbf{x} = (2, 1, 2, 6, 6, 5)$, then $S(\mathbf{x}) = 4$. Thus, $S(\mathbf{X}^L)$ is a random variable with distribution dependent on $p_X(x)$ and L.

3.3 Background

This section presents the simple substitution cryptosystem, and the erasure wiretap channel in turn.

3.3.1 Simple Substitution Cipher

Let $\mathcal{M} = \{1, 2, \ldots, N\}$ be a finite set representing the source alphabet. The message source M is memoryless, i.e., symbols are independent and identically distributed (i.i.d.), and source realizations are drawn from $\mathcal{M}$ according to the p.m.f. p_M. We will adopt the shorter notation $p_M(i) = q_i$ for $i = 1, 2, \ldots, N$, and note that p_M is not necessarily uniform over $\mathcal{M}$[1]. This signifies the lack of source coding in the system. The cryptogram alphabet $\mathcal{E}$ is equal to $\mathcal{M}$. The simple substitution encryption rule maps the message M to the cryptogram E, and is a function of the key K which is taken from the keyspace $\mathcal{K}$. Let $\mathcal{T} = \{T_j(\cdot)\}_{j=1}^{J}$ be the collection of all invertible transformations from $\mathcal{M}$ onto $\mathcal{E}$. The encryption scheme chooses a key randomly, which then acts as an index for a transformation from $\mathcal{T}$. The distribution on keys is uniform, and $\mathcal{K}$ contains every mapping of N symbols onto N symbols. Thus, $J = |\mathcal{K}| = N!$, and each occurs with probability $\frac{1}{N!}$.

We can then write $T_K(\mathbf{M}^L) = \mathbf{E}^L$ as the encryption mechanism, or more precisely,

$$\begin{aligned} T_K(M_1, M_2, \ldots, M_L) &= (T_K(M_1), T_K(M_2), \ldots, T_K(M_L)) \\ &= (E_1, E_2, \ldots, E_L). \end{aligned} \tag{17}$$

The decryption rule is similarly denoted as $T_K^{-1}(\mathbf{E}^L) = \mathbf{M}^L$. Of course, the cryptosystem is such that

$$T_k^{-1}(T_k(\mathbf{m}^L)) = \mathbf{m}^L, \forall k \in \mathcal{K} \text{ and } \forall \mathbf{m}^L \in \mathcal{M}^L. \tag{18}$$

It was noted by Shannon in [15, pg. 674] that the simple substitution cipher is a *pure* cipher, meaning that for every $i, j, k \in \mathcal{K}$ there exists some $s \in \mathcal{K}$ such that

$$T_i T_j^{-1} T_k = T_s \tag{19}$$

and every key is equally likely. Shannon then went on to define *residue classes* for pure ciphers. In essence, the set of residue classes are mutually exclusive in the message space $\mathcal{M}^L$, and the messages from a particular residue class can all be enciphered into the same subset of cryptograms. This signifies that there are residue classes in $\mathcal{E}^L$ as well as in $\mathcal{M}^L$. For the substitution cipher, the set of residue classes of $\mathcal{M}^L$ are

[1] This nonuniformity provides the structure for frequency analysis attacks against the cryptosystem by matching up the most frequent symbols of ciphertext with the most likely symbols of plaintext.

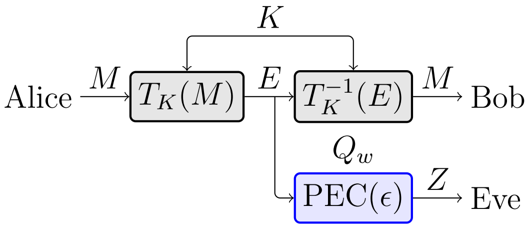

Figure 8: Packet erasure wiretap channel model for simple substitution cipher.

the set of length-L patterns that can be formed using an alphabet of size N. Also, encryption does not change the pattern of symbols in a substitution cipher, so $\mathbf{m}^L$ can only be encrypted into $\mathbf{e}^L$ if $\Psi(\mathbf{m}^L) = \Psi(\mathbf{e}^L)$.

3.3.2 Erasure Wiretap Channel

Figure 8 shows the channel model we assume throughout this chapter to be the packet erasure wiretap channel. Alice wishes to transmit a message M secretly to Bob. She thus employs a secret key K to encrypt M into a cryptogram E using the simple substitution cipher. Bob receives E error free over a noiseless communications channel, and applies his knowledge of K to decrypt E back to M. An eavesdropper named Eve listens in on the communication between Alice and Bob, albeit with independent randomly occurring packet erasures, where packets are erased with probability ϵ and obtained error free with probability $(1-\epsilon)$. For our analysis, each packet is comprised of exactly one ciphertext symbol; thus, the channel model could also be presented as a symbol erasure channel. The received vector of symbols at the eavesdropper is given as $\mathbf{Z}^L = (Z_1, Z_2, \ldots, Z_L)$, where each Z_i is a random variable over the alphabet $\mathcal{Z} = \{\mathcal{M} \cup e\}$. Note that this model is akin to the overall system model given in Figure 5 in Section 2.1, but only the pieces crucial to this chapter are shown in Figure 8.

3.4 Security with Noisy Ciphertext

The following was noted in [15] and later more formally in [39].

Theorem 1 (Dunham [39], Theorem 1)**.** *For any cipher system*

$$\mathbb{H}(\mathbf{M}^L|\mathbf{E}^L) + \mathbb{H}(K|\mathbf{M}^L, \mathbf{E}^L) = \mathbb{H}(K|\mathbf{E}^L). \tag{20}$$

Proof. The proof of the theorem is instructive. The starting point is given by equating the two relations

$$\begin{aligned} \mathbb{H}(K, \mathbf{M}^L|\mathbf{E}^L) &= \mathbb{H}(\mathbf{M}^L|\mathbf{E}^L) + \mathbb{H}(K|\mathbf{M}^L, \mathbf{E}^L) \\ \mathbb{H}(K, \mathbf{M}^L|\mathbf{E}^L) &= \mathbb{H}(K|\mathbf{E}^L) + \mathbb{H}(\mathbf{M}^L|K, \mathbf{E}^L). \end{aligned} \tag{21}$$

Then the theorem is easily shown by realizing that $\mathbb{H}(\mathbf{M}^L|K, \mathbf{E}^L) = 0$ by (18), that is, knowing the key and the cryptogram leaves no uncertainty about the message. □

Consider the case where encrypted symbols are erased independently at random with probability ϵ to form $\mathbf{Z}^L$. Equating the following expressions no longer leads us to an obvious simplification.

$$\begin{aligned} \mathbb{H}(K, \mathbf{M}^L|\mathbf{Z}^L) &= \mathbb{H}(\mathbf{M}^L|\mathbf{Z}^L) + \mathbb{H}(K|\mathbf{M}^L, \mathbf{Z}^L) \\ \mathbb{H}(K, \mathbf{M}^L|\mathbf{Z}^L) &= \mathbb{H}(K|\mathbf{Z}^L) + \mathbb{H}(\mathbf{M}^L|K, \mathbf{Z}^L). \end{aligned} \tag{22}$$

Now, only knowing the received error-prone vector $\mathbf{Z}^L$ and K, we cannot recover $\mathbf{M}^L$ unless $\mathbf{Z}^L$ has no erasures. We can, however, calculate $\mathbb{H}(\mathbf{M}^L|K, \mathbf{Z}^L)$ without too much work. Making use of the chain rule and the independent and memoryless nature of the source and the channel [13], we have the following derivation.

$$\begin{aligned}
\mathbb{H}(\mathbf{M}^L|K,\mathbf{Z}^L) &= \sum_{i=1}^{L} \mathbb{H}(M_i|K,\mathbf{Z}^L, M_{i+1},\ldots,M_L) \\
&= \sum_{i=1}^{L} \mathbb{H}(M_i|K,Z_i) \\
&= L\mathbb{H}(M|K,Z) \\
&= L\sum_{k\in\mathcal{K}}\sum_{z\in\mathcal{Z}} p_K(k)p_{Z|K}(z|k)\mathbb{H}(M|k,z) \\
&= L\sum_{k\in\mathcal{K}} p_K(k)p_{Z|K}(Z=e|k)\mathbb{H}(M|k,Z=e) \\
&= L\sum_{k\in\mathcal{K}} p_K(k)\epsilon\mathbb{H}(M) \\
&= \epsilon L\mathbb{H}(M). \qquad (23)
\end{aligned}$$

Now we combine (22) and (23), and state the following lemma.

Lemma 1. *If message symbols are i.i.d., then for any cipher system where E_i is only a function of K and M_i, and ciphertext symbols in $\mathbf{E}^L$ are then erased independently at random with probability ϵ to yield $\mathbf{Z}^L$,*

$$\mathbb{H}(K|\mathbf{Z}^L) = \mathbb{H}(\mathbf{M}^L|\mathbf{Z}^L) + \mathbb{H}(K|\mathbf{M}^L,\mathbf{Z}^L) - \epsilon L\mathbb{H}(M). \qquad (24)$$

This change in the relationship of equivocations due to the erasure channel comes as no surprise. Intuitively, we might anticipate such a term as $\epsilon L\mathbb{H}(M)$ to enter into the relationship between key and message equivocation, as the i.i.d. nature of the source symbols clearly gives no way of recovering M if $Z = e$, even if the key is known.

3.4.1 Key Equivocation

To solve for the new key equivocation given erasure-prone ciphertext, $\mathbb{H}(K|\mathbf{Z}^L)$, we first look to the noise-free solution originally solved in [38] where it was shown that

$$\mathbb{H}(K|\mathbf{E}^L) = \sum_{|\mathbf{x}|=L} \frac{L!}{x_1!x_2!\cdots x_N!} \prod_{n=1}^{N} q_n^{x_n} \log \frac{\sum_{l=1}^{N!}\prod_{n=1}^{N} q_{t_l(n)}^{x_n}}{\prod_{n=1}^{N} q_n^{x_n}}. \qquad (25)$$

The one new piece of notation in this expression is the vector $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ which is a frequency vector for messages $\mathbf{m}^L \in \mathcal{M}^L$. The ith element of $\mathbf{x}$ signifies the number of times i occurs in $\mathbf{m}^L$. For example, if $\mathbf{m}^L = (3, 2, 4, 2, 2, 3, 3, 3)$, and $N = 5$, then $\mathbf{x} = (0, 3, 4, 1, 0)$. The first sum in (25) ranges over all $\mathbf{x}$ such that

$$|\mathbf{x}| = \sum_{n=1}^{N} x_n = L. \tag{26}$$

The following relationship exists between the key equivocation under the erasure channel case and that obtained through noise-free ciphertext.

Lemma 2 (Key Equivocation)**.** *When a message of i.i.d. source symbols* $\mathbf{M}^L$ *is encrypted using a simple substitution cipher, and ciphertext symbols* $\mathbf{E}^L$ *are erased independently at random with probability* ϵ *to form* $\mathbf{Z}^L$*, then*

$$\mathbb{H}(K|\mathbf{Z}^L) = \sum_{i=0}^{L} \binom{L}{i} \epsilon^{(L-i)} (1-\epsilon)^i \mathbb{H}(K|\mathbf{E}^i). \tag{27}$$

Proof.

$$\mathbb{H}(K|\mathbf{Z}^L) = \sum_{k=1}^{N!} \sum_{\mathbf{z}^L \in \mathcal{Z}^L} p_{\mathbf{Z}^L K}(\mathbf{z}^L, k) \log \frac{\sum_{l=1}^{N!} p_{\mathbf{Z}^L K}(\mathbf{z}^L, l)}{p_{\mathbf{Z}^L K}(\mathbf{z}^L, k)}, \tag{28}$$

but

$$p_{\mathbf{Z}^L K}(\mathbf{z}^L, k) = \sum_{\mathbf{m}^L \in \mathcal{M}^L} p_{\mathbf{Z}^L | K \mathbf{M}^L}(\mathbf{z}^L | k, \mathbf{m}^L) p_K(k) p_{\mathbf{M}^L}(\mathbf{m}^L), \tag{29}$$

and

$$p_{\mathbf{Z}^L | K \mathbf{M}^L}(\mathbf{z}^L | k, \mathbf{m}^L) = \begin{cases} \epsilon^{(L-i)} (1-\epsilon)^i & \text{if } \mathbf{z}^L \overset{e}{=} T_k(\mathbf{m}^L), \\ 0 & \text{otherwise,} \end{cases} \tag{30}$$

where $\mathbf{z}^L$ has $(L-i)$ erasures. This implies that

$$p_{\mathbf{Z}^L K}(\mathbf{z}^L, k) = \frac{\epsilon^{(L-i)} (1-\epsilon)^i}{N!} \sum_{\mathbf{e}^L : \mathbf{e}^L \overset{e}{=} \mathbf{z}^L} p_{\mathbf{M}^L}(T_k^{-1}(\mathbf{e}^L)) \tag{31}$$

Without loss of generality, assume the erasures are indexed last. Then,

$$p_{\mathbf{Z}^L K}(\mathbf{z}^L, k) = \frac{\epsilon^{(L-i)} (1-\epsilon)^i}{N!} p_{\mathbf{M}^i}(T_k^{-1}(\mathbf{z}^i)). \tag{32}$$

Now, because the symbols of $\mathbf{m}^L$ are independent, we can use the frequency vector to write

$$p_{\mathbf{M}^i}(T_k^{-1}(\mathbf{z}^i)) = q_1^{x_1} q_2^{x_2} \ldots q_N^{x_N}, \tag{33}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ now represents the frequency of symbols that occur only in the non-erased symbols as they are mapped back to $\mathcal{M}$. Finally, plugging the combined result of (33) and (32) back into (28), along with some careful counting to include all the ways in which a single frequency vector can occur, yields the following expression.

$$\mathbb{H}(K|\mathbf{Z}^L) = \sum_{i=0}^{L} \binom{L}{i} \epsilon^{(L-i)} (1-\epsilon)^i \sum_{|\mathbf{x}|=i} \frac{i!}{x_1! x_2! \ldots x_N!} \prod_{n=1}^{N} q_n^{x_n} \log \frac{\sum_{l=1}^{N!} \prod_{n=1}^{N} q_{t_l(n)}^{x_n}}{\prod_{n=1}^{N} q_n^{x_n}}. \tag{34}$$

Comparison of (34) and (25) shows the proof to be complete. □

3.4.2 Message Equivocation

In solving for the result of the message equivocation $\mathbb{H}(\mathbf{M}^L|\mathbf{Z}^L)$, we first remark that knowing $\mathbb{H}(K|\mathbf{Z}^L)$ allows us to solve for the key appearance equivocation[2] $\mathbb{H}(K|\mathbf{M}^L\mathbf{Z}^L)$, and then apply the result from Lemma 1 given in (24) to solve for the message equivocation as

$$\mathbb{H}(\mathbf{M}^L|\mathbf{Z}^L) = \epsilon L \mathbb{H}(M) + \mathbb{H}(K|\mathbf{Z}^L) - \mathbb{H}(K|\mathbf{M}^L, \mathbf{Z}^L). \tag{35}$$

Since $\mathbb{H}(K|\mathbf{M}^L, \mathbf{Z}^L)$ is a much simpler calculation than $\mathbb{H}(\mathbf{M}^L|\mathbf{Z}^L)$, we will take this approach.

Again, we start by borrowing a result about the key appearance equivocation for noiseless ciphertext.

Theorem 2 (Dunham [39], Theorem 2)**.** *If a discrete source is enciphered by a simple substitution cipher with equiprobable key and the source alphabet has N letters, we*

[2] The expression *key appearance equivocation* was originally given to the quantity $\mathbb{H}(K|\mathbf{M}^L, \mathbf{E}^L)$ to signify the remaining key equivocation following an L-length message in a known plaintext attack [39]. The same name is used to describe $\mathbb{H}(K|\mathbf{M}^L, \mathbf{Z}^L)$ here.

have

$$\mathbb{H}(K|\mathbf{M}^L, \mathbf{E}^L) = \sum_{i=0}^{N} P_{S^L}(i) \log((N-i)!). \tag{36}$$

Recall that $S(\mathbf{M}^L)$ signifies the number of unique symbols in $\mathbf{M}^L$. The notation $P_{S^L}(i)$ is simply the probability that $S(\mathbf{M}^L)$ equals i. We will consider varying length vectors in the following derivations, and so more generally, $P_{S^j}(i) = \Pr(S(\mathbf{M}^j) = i)$ for any length-j vector of symbols.

Lemma 3 (Key Appearance Equivocation). *If symbols in $\mathbf{M}^L$ are i.i.d. and encrypted using a simple substitution cipher with equally likely keys to form $\mathbf{E}^L$, and then $\mathbf{Z}^L$ is formed by erasing symbols from $\mathbf{E}^L$ independently with probability ϵ, then*

$$\mathbb{H}(K|\mathbf{M}^L, \mathbf{Z}^L) = \sum_{i=0}^{L} \binom{L}{i} \epsilon^{L-i}(1-\epsilon)^i \mathbb{H}(K|\mathbf{M}^i, \mathbf{E}^i). \tag{37}$$

Proof.

$$\begin{aligned}\mathbb{H}(K|\mathbf{M}^L, \mathbf{Z}^L) &= \sum_{\mathbf{m}^L \in \mathcal{M}^L} \sum_{\mathbf{z}^L \in \mathcal{Z}^L} p_{\mathbf{M}^L\mathbf{Z}^L}(\mathbf{m}^L, \mathbf{z}^L) \mathbb{H}(K|\mathbf{M}^L = \mathbf{m}^L, \mathbf{Z}^L = \mathbf{z}^L) \\ &= \sum_{\mathbf{m}^L \in \mathcal{M}^L} \sum_{\mathbf{z}^L \in \mathcal{Z}^L} p_{\mathbf{M}^L\mathbf{Z}^L}(\mathbf{m}^L, \mathbf{z}^L) \log((N - S(\mathbf{z}^L))!) \end{aligned} \tag{38}$$

because there are exactly $S(\mathbf{z}^L)$ unique symbols that are revealed in $\mathbf{z}^L$. Given we know the plaintext as well in the key appearance equivocation, this leaves $N - S(\mathbf{z}^L)$ symbols that we have yet to identify in the key. Thus,

$$\mathbb{H}(K|\mathbf{M}^L = \mathbf{m}^L, \mathbf{Z}^L = \mathbf{z}^L) = \log((N - S(\mathbf{z}^L))!) \tag{39}$$

because there are $(N - S(\mathbf{z}^L))!$ equally likely keys that are possible. We can also write

$$\begin{aligned} p_{\mathbf{M}^L\mathbf{Z}^L}(\mathbf{m}^L, \mathbf{z}^L) &= \sum_{k \in \mathcal{K}} p_{\mathbf{M}^L\mathbf{Z}^L K}(\mathbf{m}^L, \mathbf{z}^L, k) \\ &= \sum_{k=1}^{N!} p_{\mathbf{Z}^L|\mathbf{M}^L K}(\mathbf{z}^L|\mathbf{m}^L, k) p_{\mathbf{M}^L}(\mathbf{m}^L) p_K(k) \\ &= \frac{p_{\mathbf{M}^L}(\mathbf{m}^L)}{N!} \sum_{k=1}^{N!} p_{\mathbf{Z}^L|\mathbf{M}^L K}(\mathbf{z}^L|\mathbf{m}^L, k). \end{aligned} \tag{40}$$

Now, assume that $\mathbf{z}^L$ has $(L-i)$ erasures,

$$p_{\mathbf{Z}^L|\mathbf{M}^L K}(\mathbf{z}^L|\mathbf{m}^L,k) = \begin{cases} \epsilon^{(L-i)}(1-\epsilon)^i & \text{if } \mathbf{z}^L \overset{e}{=} T_k(\mathbf{m}^L), \\ 0 & \text{otherwise.} \end{cases} \tag{41}$$

There are exactly $(N - S(\mathbf{z}^L))!$ keys that will exhibit a nonzero value in (41). Thus,

$$p_{\mathbf{M}^L\mathbf{Z}^L}(\mathbf{m}^L,\mathbf{z}^L) = \begin{cases} \frac{(N-S(\mathbf{z}^L))!}{N!} p_{\mathbf{M}^L}(\mathbf{m}^L)\epsilon^{(L-i)}(1-\epsilon)^i & \text{if } \Psi(\mathbf{m}^L) \overset{e}{=} \Psi(\mathbf{z}^L) \\ 0 & \text{otherwise.} \end{cases} \tag{42}$$

Combining the expressions in (38) and (42) yields

$$\begin{aligned}\mathbb{H}(K|\mathbf{M}^L,\mathbf{Z}^L) = \sum_{\mathbf{m}^L\in\mathcal{M}^L} \sum_{\mathbf{z}^L:\Psi(\mathbf{z}^L)\overset{e}{=}\Psi(\mathbf{m}^L)} \frac{(N-S(\mathbf{z}^L))!}{N!} \times \\ \times p_{\mathbf{M}^L}(\mathbf{m}^L)\epsilon^{(L-i)}(1-\epsilon)^i \log((N-S(\mathbf{z}^L))!).\end{aligned} \tag{43}$$

Since we are summing over all possible messages in $\mathcal{M}^L$, and because both erasures and source symbols are independent, we may fix the erasure pattern and simplify the expression so that

$$\begin{aligned}\mathbb{H}(K|\mathbf{M}^L,\mathbf{Z}^L) = \sum_{\mathbf{m}^L\in\mathcal{M}^L} p_{\mathbf{M}^L}(\mathbf{m}^L) \sum_{i=0}^{L} \binom{L}{i} \epsilon^{(L-i)}(1-\epsilon)^i \times \\ \times \sum_{\mathbf{e}^i:\Psi(\mathbf{e}^i)=\Psi(\mathbf{m}^i)} \frac{(N-S(\mathbf{e}^i))!}{N!} \log((N-S(\mathbf{e}^i))!).\end{aligned} \tag{44}$$

Furthermore, since there are exactly $\frac{N!}{(N-S(\mathbf{e}^i))!}$ different symbol combinations for $\mathbf{e}^i$ in the residue class of $\mathbf{m}^i$, and because $S(\mathbf{e}^i) = S(\mathbf{m}^i)$, therefore, we gain a further simplification so that

$$\mathbb{H}(K|\mathbf{M}^L,\mathbf{Z}^L) = \sum_{i=0}^{L} \binom{L}{i} \epsilon^{(L-i)}(1-\epsilon)^i \sum_{\mathbf{m}^L\in\mathcal{M}^L} p_{\mathbf{M}^L}(\mathbf{m}^L) \log((N-S(\mathbf{m}^i))!). \tag{45}$$

Thus we can group elements in the sum according to $S(\mathbf{m}^i)$, and write the final expression

$$\mathbb{H}(K|\mathbf{M}^L,\mathbf{Z}^L) = \sum_{i=0}^{L} \binom{L}{i} \epsilon^{(L-i)}(1-\epsilon)^i \sum_{j=0}^{N} P_{S^i}(j) \log((N-j)!). \tag{46}$$

Finally, by comparison of (46) and (36), we arrive at (37), and the proof is complete.

□

We can now combine our results to write the expression for the message equivocation.

Theorem 3 (Message Equivocation)**.** *For a simple substitution cipher that encrypts an L-length memoryless plaintext message* $\mathbf{M}^L$ *into ciphertext* $\mathbf{E}^L$*, if symbols are erased independently with probability* ϵ *from* $\mathbf{E}^L$ *to form* $\mathbf{Z}^L$*, then*

$$\mathbb{H}(\mathbf{M}^L|\mathbf{Z}^L) = \epsilon L\mathbb{H}(M) + \sum_{i=0}^{L} \binom{L}{i} \epsilon^{(L-i)}(1-\epsilon)^i \mathbb{H}(\mathbf{M}^i|\mathbf{E}^i). \tag{47}$$

Proof. From (24) in Lemma 1, we receive our starting point, and thus can derive the relationship as follows.

$$\begin{aligned}\mathbb{H}(\mathbf{M}^L|\mathbf{Z}^L) &= \mathbb{H}(K|\mathbf{Z}^L) - \mathbb{H}(K|\mathbf{M}^L, \mathbf{Z}^L) + \epsilon L\mathbb{H}(M) \\ &= \epsilon L\mathbb{H}(M) \sum_{i=0}^{L} \binom{L}{i} \epsilon^{(L-i)}(1-\epsilon)^i \times \\ &\qquad\qquad \times \left[\mathbb{H}(K|\mathbf{E}^i) - \mathbb{H}(K|\mathbf{M}^i, \mathbf{E}^i)\right] \\ &= \epsilon L\mathbb{H}(M) \sum_{i=0}^{L} \binom{L}{i} \epsilon^{(L-i)}(1-\epsilon)^i \mathbb{H}(\mathbf{M}^i|\mathbf{E}^i). \qquad (48)\end{aligned}$$

The second relation comes directly from (27) in Lemma 2 and (37) in Lemma 3, while the final expression comes from (20) in Theorem 1. □

3.5 Discussion and Conclusions

In conclusion, we have solved for the key equivocation, key appearance equivocation, and message equivocation when cryptogram symbols are obtained independently through a symbol erasure channel by an eavesdropper. We have chosen to discuss the respective results of Lemma 2, Lemma 3, and Theorem 3 together because of the similar nature of the expressions in (27), (37), and (47). These expressions show that the three equivocations are all affected similarly by the erasure channel. We see a weighted sum in each expression, where the total amount of equivocation is equal to the combination of equivocations for clean cryptograms of all lengths between zero and L. The weighting of each equivocation is simply the probability that the right number of erasures will occur to leave an effective cryptogram of that length. Only the message equivocation carries an additional term signifying the lost information that cannot be recovered about the message due to erasures. This additional term also presents a bias in the equivocation such that the message equivocation will only converge to zero if no erasures occur in the channel. Therefore, the key equivocation should be used to measure the strength of the system rather than the message equivocation.

These relations are clearly dependent on the memoryless message source in the problem setup, and prompt us to consider message sources with additional structure in the future so as to apply the results to encrypted messages in a true spoken language. We see also, through simple comparisons of the results with their erasure-free counterparts, that if a wiretap code can force erasures of ciphertext symbols upon an eavesdropper, that the total equivocation at the eavesdropper will increase. Surely this also prompts the continued study of the interplay between physical-layer security and cryptography.

CHAPTER IV

CRYPTANALYSIS OF STREAM CIPHER WITH BIT ERRORS

Chapter 3 dealt with the information-theoretic cryptanalysis of a classic cipher when symbols of ciphertext are erased in the wiretap channel. Although the techniques may lead to greater understanding of other more modern ciphers in a noisy ciphertext setting, we also wish to deal with more practical scenarios directly. In this chapter, we look into the cryptanalysis of stream ciphers, and evaluate the enhancement to security that can be gained when the ciphertext is error prone by analyzing specific attack algorithms. The stream ciphers that we investigate here have a keystream generator that is based on linear-feedback shift registers (LFSRs). It is well known that this class of ciphers is susceptible to certain correlation attacks [42] that exploit the linear structure of the keystream along with correlations within the keystream generator. However, we wish to characterize the security of these ciphers when the ciphertext obtained by an attacker is error prone. We show that two classes of correlation attacks can still be implemented with essentially no changes to the attack algorithms in this setting, although with limited effectiveness. The degeneration of the potency of these attacks is given as a function of the error rate in the ciphertext, and is characterized using computational security. Of course, as we gain understanding of the required error rates in ciphertext to cause attacks against stream ciphers to fail, we can use this knowledge to design practical physical-layer security codes. The addition of such codes has the ability to make weak cryptosystems strong, and strong cryptosystems even stronger, by increasing the entropy of the attacker's knowledge of the ciphertext E. The material in this chapter is drawn from the published works in [6, 7, 8].

4.1 Background

Prior to the introductory discussion on stream ciphers, we must first present the binary symmetric channel, a metric called the Hamming distance, and a basic building block of pseudorandom sequences called the linear-feedback shift register (LFSR).

4.1.1 Binary Symmetric Channel

The binary symmetric channel (BSC) is perhaps the simplest channel model that takes errors into account. The model is depicted in Figure 9. Here we see that bits at the input are flipped by the channel with probability p, and transmitted error free with probability $(1-p)$. Although this model seems simplistic, it actually captures the essence of many real channels when hard bit decisions are made at the receiver [13]. The BSC will be used to model correlations in bit streams, and error rates across channels in this chapter.

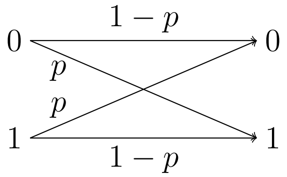

Figure 9: Binary symmetric channel model.

4.1.2 Hamming Distance

The following definition is borrowed from [12].

Definition 10. The *Hamming distance* between a sequence $\mathbf{x}^n = (x_1, x_2, \ldots, x_n)$ and a sequence $\mathbf{y}^n = (y_1, y_2, \ldots, y_n)$ is the number of positions where the corresponding elements differ:

$$d_H(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} [x_i \neq y_i], \tag{49}$$

where

$$[x_i \neq y_i] = \begin{cases} 1 & \text{if } x_i \neq y_i \\ 0 & \text{if } x_i = y_i. \end{cases}$$

The notion of Hamming distance will be important when we start to discuss attack strategies for stream ciphers.

4.1.3 Linear-Feedback Shift Registers

An LFSR is nothing more than a binary shift register that produces an output bit with each shift of the register. To add a pseudorandom quality to the output of the register, bits in the register can either be fed back or fed forward with connections to specific bit slots in the register. For this chapter, we will assume that connections feed back into the register as shown in Figure 10. The feedback connections of an LFSR can be specified in polynomial form using a *connection polynomial.* The polynomial takes the form of

$$g(x) = g_0 + g_1 x + g_2 x^2 + \cdots + g_\nu x^\nu, \tag{50}$$

where ν is the order of the polynomial (also the length of the associating LFSR) and $g_j \in \{0,1\}$ for $j = 0, 1, \ldots, \nu$. Feedback connections appear only at those locations in the register where $g_j = 1$. Thus, the connection polynomial for the LFSR in Figure 10 is $g(x) = 1 + x + x^4$.

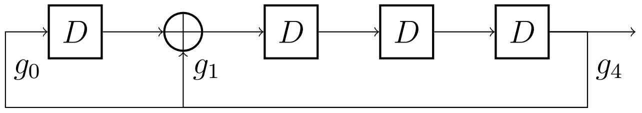

Figure 10: Linear-feedback shift register (LFSR) with connection polynomial $g(x) = 1 + x + x^4$, with nonzero feedback coefficients labeled.

The D blocks are delays in the register, and can be implemented in practice using D flip flops. Typically, $g(x)$ in a cryptographic keystream generator is chosen such that the pseudorandom output sequence is of maximal length before repeating. Consider the state of an LFSR to be defined by the contents of the register. Since each position in the register can hold either a zero or a one, we may jump to the conclusion that a maximal-length output sequence would be one that would traverse all possible 2^{ν} states of the LFSR. However, the all-zero state causes the LFSR to always output zeros, and therefore, is not part of a maximal-length sequence. Thus, an LFSR output sequence can be at most $2^{\nu} - 1$ bits long before repeating. A connection polynomial that produces a maximal-length sequence is a *primitive* polynomial over $GF(2)$, that is, the Galois field with two elements (zero and one). For more on this subject, the interested reader is referred to [12] or a suitable textbook on abstract algebra. The connection polynomial of the LFSR in Figure 10 is primitive.

4.1.4 LFSR-Based Stream Ciphers

During the 1970s, stream ciphers were introduced as an approximation to the one-time pad. Initially, they were thought to be quite secure. After all, it was already known, thanks to Shannon, that the one-time pad could offer *perfect secrecy* as was outlined in Section 2.3.1. The encryption technique takes as inputs the length-n binary message $\mathbf{M}^n = (M_1, M_2, \ldots, M_n)$, and a keystream $\mathbf{K}^n = (K_1, K_2, \ldots, K_n)$, and calculates the ciphertext as

$$E_i = M_i \oplus K_i \tag{51}$$

for $i = 1, 2, \ldots, n$, to form $\mathbf{E}^n = (E_1, E_2, \ldots, E_n)$, where $\oplus$ signifies the XOR operation. Note this is identical to (3), the one-time pad encryption rule. The decryption rule is symmetric to the encryption rule, just as in the one-time pad. Bob also has access to $\mathbf{K}^n$, and then calculates

$$M_i = E_i \oplus K_i \tag{52}$$

for $i = 1, 2, \ldots, n$. The difference between the one-time pad and a stream cipher of the 1970's can be seen in the keystream $\mathbf{K}^n$. The one-time pad required that $\mathbf{K}^n$ be uniformly random, whereas the more practical stream cipher requires only a random seed, and then generates a pseudorandom sequence $\mathbf{K}^n$ from that seed. The stream ciphers we will analyze in this chapter have a common keystream generator structure that is pictured in Figure 11. The keystream $\mathbf{K}^n$ is generated using β distinct LFSR output sequences as inputs to some combining function $f(\cdot)$. Examples of some well-studied combining functions are given in [43, 44].

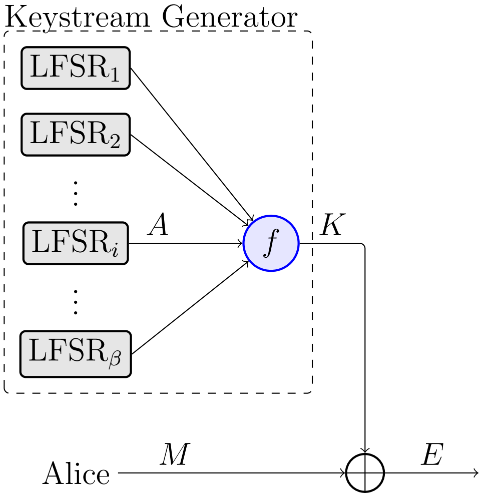

Figure 11: Keystream generator based on linear-feedback shift registers.

The true secret key of the cipher is comprised of the initial contents of each of the shift registers, and sometimes the connection polynomials of the LFSRs as well. It was shown that if the output streams of the LFSRs were combined in a linear fashion, the key could be found quite easily assuming a small amount of known plaintext, using e.g. the Berlekamp-Massey shift register synthesis algorithm [45]. However, even using non-linear combining functions, the output sequence $\mathbf{A}^n$ of a single LFSR, say the ith one, and the keystream generator output $\mathbf{K}^n$ might be correlated in such a system [46]. This correlation was first exploited in an attack on the cipher by Siegenthaler in 1985 using only ciphertext [47]. The result was an attack that had drastically reduced complexity compared to a brute-force attack. Known plaintext attacks soon followed with even faster attack algorithms that were duly named *fast correlation* *attacks*. The first two such algorithms were published by Meier and Staffelbach in 1989 [48]. Others followed including [49] and [50]. These attacks commonly treat only the initial contents of the LFSRs as the secret key and assume that the connection polynomials for all LFSRs are common knowledge. Although Siegenthaler produced the first correlation attack, he also strengthened stream ciphers of this nature by developing the notion of correlation-immune combining functions [42]. Additional criteria in the design of $f(\cdot)$ that reduce the effectiveness of fast correlation attacks are provided in [48].

Clearly much is known regarding this family of ciphers, and attacks against them are well developed. Therefore, LFSR-based stream ciphers provide an ideal backdrop on which to cast the notion of physical-layer security as a cryptographic enhancement. We will take a similar approach as in Chapter 3 to modeling the effects of physical-layer security coding, assuming a positive error rate over the wiretap channel Q_w. The main channel Q_m will once again be assumed to be noiseless for the sake of the analysis. It should be noted that this problem could potentially be cast as a learning parity with noise (LPN) problem. The interested reader is referred to the following sources for more information [51, 52, 53].

We also adjust the cryptanalysis to a slightly more practical methodology. Rather than solving for equivocations of message and key, here we choose to evaluate known attacks against stream ciphers. It will be shown that LFSR-based stream ciphers, although generally susceptible to certain fast correlation attacks, can be made secure against two classes of known attacks through exploiting noise in an eavesdropper's channel, or more precisely, we show the required error rate in Q_w to reduce specific fast correlation attacks to brute force attacks, thus removing any advantage an attacker may derive from correlation. This requires an understanding of specific fast correlation attacks.

4.1.5 Outline of Chapter

Regarding the remainder of the chapter, in Section 4.2 we provide the details of two fast correlation attacks that were among the earliest of the fast-correlation variety. It will then be shown how to analyze the effectiveness of these attacks when allowing for the possibility of corrupt ciphertext in Section 4.3. Oddly, we will see that the concept of noisy ciphertext changes only a single parameter in the noise-free ciphertext cryptanalysis. The Attack 2 algorithm reveals the secret key, or initial contents of a shift register, iteratively. For this attack, we apply the tool of extrinsic information transfer (EXIT) charts to show the expected attack progression and outcome. EXIT charts were designed to give a graphical understanding and portrayal of iterative decoders in turbo codes [54, 55], and have been used to provide insight on low-density parity-check (LDPC) decoding as well [12]. The authors of [56] show deeper analysis on the topic of EXIT curves and expound on information-theoretical implications of EXIT charts. For our purposes, EXIT chart analysis gives a clear indication when attacks are expected to fail and when they are expected to succeed as a function of system and channel parameters. We provide two ways of forming EXIT curves to chart the progression of Attack 2 in Sections 4.4 and 4.5, using only hard decision information, and then using soft decision information, respectively. We finally summarize the findings of the chapter in Section 4.6

4.2 Correlation Attacks

The main assumption of fast correlation attacks on LFSR-based stream ciphers is that the encryption technique shown in Figure 11 can be modeled as a single LFSR output sequence $\mathbf{A}^n$ with a following BSC with crossover probability p_1 to produce the keystream $\mathbf{K}^n$. This simplified keystream generator model is depicted in Figure 12, and implies that $\mathbf{A}^n$ and $\mathbf{K}^n$ are correlated such that $\Pr(A_j = K_j) = 1 - p_1 \forall j$.

Using this model, fast correlation attacks attempt to retrieve the initial contents

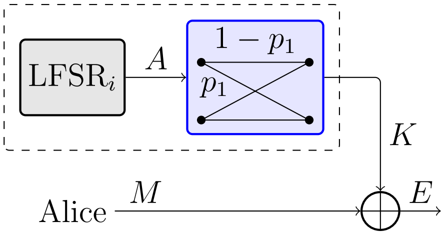

Figure 12: Simplified model of the keystream generator assumed in fast correlation attacks.

of each of the β distinct LFSRs in the keystream generator one at a time. The attacks rely on the structure of $\mathbf{A}^n$, and thus can be applied, although imperfectly, to $\mathbf{K}^n$.[1] Recall that $\mathbf{A}^n$ is not truly random, but rather pseudorandom. Checksums, or simply *checks*, that exist in $\mathbf{A}^n$ are explicit in the connection polynomial $g(x)$ of its associating LFSR. Let t be one less than the number of nonzero coefficients in $g(x)$, and denote the indices of the nonzero coefficients as $j_0, j_1, \ldots, j_t$. Then the following expression holds in $\mathbf{A}^n$ for any j as long as all the indices are less than n

$$A_{j+j_0} + A_{j+j_1} + \cdots + A_{j+j_t} = 0. \tag{53}$$

We can solve this expression for any single bit in $\mathbf{A}^n$ as

$$A_{j+j_u} = A_{j+j_0} + A_{j+j_1} + \cdots + A_{j+j_{u-1}} + A_{j+j_{u+1}} + \cdots + A_{j+j_t}. \tag{54}$$

Almost every bit in $\mathbf{A}^n$ takes part in $t+1$ checks of this kind. Only the bits on the ends of the sequence appear in less.

Let us also take the connection polynomial and square the entire expression. The structure of the LFSR is such that the resulting polynomial $g^2(x)$ will also express a check that holds in $\mathbf{A}^n$. In fact, we can take successive square of $g(x)$ to form as many checks as possible until we overrun the length of the datastream available n [48]. One may initially feel that squaring these polynomials will prove a cumbersome

[1] Note that the idea of noisy ciphertext is akin to this attack technique where the keystream $\mathbf{K}^n$ is assumed to be a noisy version of $\mathbf{A}^n$.

task, but since these squaring operation are completed in $GF(2)$, we may use the rule of freshman exponentiation, given in the following theorem.

Theorem 4 (Theorem 5.15 in [12]). *If x and y are elements in a field of characteristic p,*

$$(x+y)^p = x^p + y^p.$$

This applies as well to polynomials with coefficients over a field of characteristic p, where the *characteristic* of a field is the smallest number of ones that add to zero in the field. In $GF(2)$, $p = 2$, and squares can, therefore, be easily calculated to provide additional checksums in $\mathbf{A}^n$.

Example 1. Let $g(x) = 1 + x + x^4$. Then $g^2(x) = 1 + x^2 + x^8$, $(g^2(x))^2 = g^4(x) = 1+x^4+x^{16}$, *etc.* These three expressions denote the following checksums in the output sequence $\mathbf{A}^n$ of the associating LFSR: $A_j + A_{j+1} + A_{j+4} = 0$, $A_j + A_{j+2} + A_{j+8} = 0$, and $A_j + A_{j+4} + A_{j+16} = 0$. Each check can be applied to any j, so long as the highest index in a check is less than n, the length of the output sequence.

Define w as the total number of checks involving the bit A_j, and enumerate these checks from one to w. We now apply check expressions to bits in $\mathbf{K}^n$. Let the vth check be

$$K_j = K_{v_1} + K_{v_2} + \cdots + K_{v_t}. \tag{55}$$

We group the bits in the right hand side of this expression as

$$B_v = \sum_{i=1}^{t} K_{v_i} \tag{56}$$

and define

$$L_v = K_j + B_v. \tag{57}$$

Then if (55) holds, $L_v = 0$. Further define

$$S = \Pr\left(B_v = \sum_{i=1}^{t} A_{v_i}\right), \tag{58}$$

which is the probability of an even number of bit flips in the bits $\{K_{v_1}, K_{v_2}, \ldots, K_{v_t}\}$. It can be shown that $S = S(t-1)$ in the recursive calculation

$$S(i) = (1-p_1)S(i-1) + p_1(1 - S(i-1)) \tag{59}$$

where $S(0) = 1 - p_1$. Now suppose that exactly h of the w checks involving A_j hold in $\mathbf{K}^n$. Without loss of generality, let checks enumerated one through h hold. Then we can define

$$\begin{aligned} P_j^* &= \Pr\left(K_j = A_j | L_1 = \cdots = L_h = 0, L_{h+1} = \cdots = L_w = 1\right) \\ &= \frac{(1-p_1)S^h(1-S)^{w-h}}{(1-p_1)S^h(1-S)^{w-h} + p_1(1-S)^h S^{w-h}}. \end{aligned} \tag{60}$$

Therefore, P_j^* is the probability that K_j equals A_j given all we know about the checksums involving K_j.

Since attacks on this system are known plaintext attacks, we will now use n to signify the number of bits in $\mathbf{M}$ known to the attacker. Although eventually we will add a physical-layer component to this system, let us assume for now that an attacker has the ciphertext $E_i = M_i \oplus K_i$ exactly for $i = 1, 2, \ldots, n$. The attacker then calculates

$$K_i = E_i \oplus M_i \tag{61}$$

for $i = 1, 2, \ldots, n$ to obtain $\mathbf{K}^n$. Performing the calculation in (60) for these n bits in $\mathbf{K}^n$, we form the vector $(\mathbf{P}^*)^n = (P_1^*, P_2^*, \ldots, P_n^*)$ [48]. The two fast correlation known-plaintext attacks are now briefly summarized. For additional details, please see [48].

4.2.1 Attack 1

The first attack from [48] is noniterative and was motivated by the intuitive notion that those bits in the keystream $\mathbf{K}^n$ that are included in the greatest number of correct checks are more likely to be equal to their corresponding bits in $\mathbf{A}^n$. Consider that each bit in $\mathbf{A}^n$ is a linear combination of the bit values in the initial state of the LFSR, which make up the part of the secret key allocated to the ith LFSR. Therefore, it is possible to solve for this portion of the secret key using ν bits with linearly independent secret key bit combinations.

An attacker selects the ν most reliable bits from $\mathbf{K}^n$, i.e., the ν bits with the highest corresponding values in $(\mathbf{P}^*)^n$, that form a linearly independent system of equations. Of course if one or more of these ν bits are in error, then the system of equations will not return the secret key as its solution. The correctness of the solution can be determined by the attacker using a threshold comparison with a correlation metric [47]. In essence this means testing out the proposed initial conditions to recover an estimate of $\mathbf{A}^n$, say $\hat{\mathbf{A}}^n$. Then the Hamming distance

$$d_H(\hat{\mathbf{A}}^n, \mathbf{K}^n) \tag{62}$$

should indicate a rough percentage of p_1 bits differing between these two sequences. It is actually a trivial matter for the attacker to estimate the channel parameter p_1 by counting correct checks in $\mathbf{K}^n$. For our analysis, we wish to consider the worst case for the legitimate parties by assuming the attacker is always able to determine whether the key obtained is correct or incorrect. If the key is incorrect, then the values of the ν bits with the highest values in $(\mathbf{P}^*)^n$ are toggled trying alternate bit sequences in order of ascending Hamming distance to the original guess until the correct key is obtained.

4.2.2 Attack 2

The second attack given in [48] calculates iterative updates between S and $(\mathbf{P}^*)^n$, and employs two nested levels of iteration. In a particular *round* of the attack, the algorithm performs multiple *iterations*. The update calculation for S in (59) is made unique for each bit-check combination. There are roughly w checks for each of n bits, but $t+1$ bits are in each check, so we can anticipate roughly

$$w' = \frac{wn}{t+1} \tag{63}$$

total checks with $t+1$ bits in each check. Hence, we construct a matrix $\mathbf{S}^{(t+1)\times w'}$ to store these values. Consider the check in (54) and call it the vth check. Let $(q_0, q_1, \ldots, q_{t-1}) = (p^*_{j+j_0}, p^*_{j+j_1}, \ldots, p^*_{j+j_{u-1}}, p^*_{j+j_{u+1}}, \ldots, p^*_{j+j_t})$, respectively. Then the value in $\mathbf{S}^{(t+1)\times w'}$ corresponding to the uth bit of the vth check is $S_{u,v}(t-1)$ and is calculated recursively as

$$S_{u,v}(i) = q_i S_{u,v}(i-1) + (1-q_i)(1-S_{u,v}(i-1)) \tag{64}$$

where $S_{u,v}(0) = q_0$. Prior to iteration p^*_{thr} and n_{thr} are calculated to act as decision thresholds. The calculations are based on an optimization of expected correction in the first iteration of the first round.

Each iteration computes $\mathbf{S}^{(t+1)\times w'}$ and $(\mathbf{P}^*)^n$ using (64) and (60) respectively, although care must be taken in applying (60), as specific values from $\mathbf{S}^{(t+1)\times w'}$ must be incorporated into the calculation.[2] The first calculation of $\mathbf{S}^{(t+1)\times w'}$ assumes $P^*_j = 1 - p_1$ for $j = 1, 2, \ldots, n$. Again, an attacker can estimate the channel parameter p_1 by counting correct checks in $\mathbf{E}^n$, as previously mentioned in Section 4.2.1. If after an iteration there are greater than n_{thr} elements of $(\mathbf{P}^*)^n$ such that $P^*_j < p_{thr}$, then the round is terminated. A round consists of a maximum of α iterations. At the end of the round, all bits K_j with corresponding $P^*_j < p_{thr}$, are flipped. All values in $(\mathbf{P}^*)^n$ are then reset to $1-p_1$, and the attack proceeds again with new calculations in $\mathbf{S}^{(t+1)\times w'}$. The attack proceeds in this fashion until it either stagnates, or converges to the correct solution.[3]

[2] Perhaps a helpful mental picture would be that of a Tanner graph in a belief propagation decoding scheme. Certain values in $\mathbf{S}^{(t+1)\times w'}$ pertain to certain bits $(\mathbf{P}^*)^n$ values, and the graph can be built so that only neighboring nodes can affect a calculation for another node.

[3] This attack has many similarities to Gallager's LDPC decoding message-passing, or belief propagation, algorithm [23].

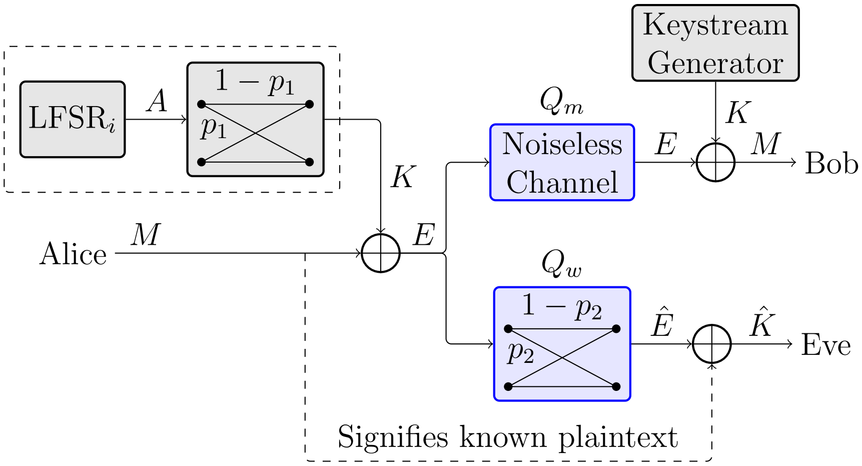

Figure 13: System model depicting a known plaintext attack against the LFSR-based stream cipher system when physical-layer security coding maintains a nonzero BER p_2 in Eve's ciphertext $\hat{E}$.

4.3 Noisy Ciphertext Analysis

For this section, we assume the system model shown in Figure 13, where the encoder, decoder, and feedback in Figure 5 are assumed to provide an effectively noiseless main channel and a positive error rate in the wiretap channel. The error rate in Q_w is modeled in Figure 13 as a BSC with probability of a bit flip p_2. Of course this implies hard decision output from Eve's decoder, and also that errors occur independently in Eve's noisy ciphertext $\hat{\mathbf{E}}^n$.

The cipher in Figure 13 is the stream cipher with an LFSR-based keystream generator presented previously in Figure 11, where encryption and decryption rules were already given in (51) and (52), respectively. Note, we have assumed the simplified model of this keystream generator from Figure 12, and therefore, the output sequence of the ith LFSR $\mathbf{A}^n$ is assumed to be correlated with the keystream $\mathbf{K}^n$ such that

$$\Pr(A_i = K_i) = 1 - p_1 \tag{65}$$

for $i = 1, 2, \ldots, n$. Bob has access to the keystream $\mathbf{K}^n$ through an identical keystream generator as the one maintained by Alice, and therefore, since Q_m is noiseless, he is able to perform the symmetric decryption operation without error to obtain $\mathbf{M}^n$. Since Eve does not initially possess $\mathbf{K}^n$, she must attack the cipher. Physical-layer security codes are assumed to render her stricken with an average bit error rate (BER) of p_2 in her ciphertext $\hat{\mathbf{E}}^n$; however, the figure also shows that Eve has access to some plaintext. We assume that exactly n bits of plaintext are known to her, and hence treat all data sequences as length-n sequences in the known plaintext attacks from Sections 4.2.1 and 4.2.2.

The attacks in Sections 4.2.1 and 4.2.2, originally from [48], exhibit a convenient property for noisy ciphertext cryptanalysis. We can actually address the notion of noisy ciphertext within the current parameters of the attacks with no change to the attack algorithms. It should be noted once again that the goal of these attacks is to recover the secret key, which is comprised of the initial contents of each LFSR in the keystream generator. The initial contents of the ith LFSR are trivial to deduce from the complete sequence $\mathbf{A}^n$.

Let us assume that the BSC in Q_w produces an error sequence $\mathbf{N}^n$, where $N_i = 1$ when a bit is flipped in the sequence. Then

$$\hat{E}_i = E_i \oplus N_i \tag{66}$$

for $i = 1, 2, \ldots, n$. When Eve applies her knowledge of the plaintext to her noisy ciphertext $\hat{\mathbf{E}}^n$, she calculates

$$\begin{aligned} \hat{E}_i \oplus M_i &= (K_i \oplus M_i \oplus N_i) \oplus M_i \\ &= K_i \oplus N_i. \end{aligned} \tag{67}$$

Clearly this result indicates that the error rate in Q_w serves to confuse Eve about $\mathbf{K}^n$ when she has access to the plaintext. Let us call Eve's noisy keystream

$$\hat{\mathbf{K}}^n = \mathbf{K}^n \oplus \mathbf{N}^n, \tag{68}$$

where, of course, the XOR operation is performed bit-wise. Then Figure 14 shows the effective progression of data sequences from $\mathbf{A}^n$ to $\hat{\mathbf{K}}^n$. The keystream $\mathbf{K}^n$ is obtained from $\mathbf{A}^n$ through a BSC where $\Pr(K_j \neq A_j) = p_1$, and the noisy keystream sequence $\hat{\mathbf{K}}^n$ is obtained by $\mathbf{K}^n$ through a BSC where $\Pr(\hat{K}_j \neq K_j) = p_2$. Thus, a further simplification can occur in the modeling of the relationship between $\mathbf{A}^n$ and $\hat{\mathbf{K}}^n$ by combining the two cascaded BSCs into a single BSC with $\Pr(\hat{K}_j \neq A_j) = p'$. This model is shown in Figure 15, and is coincidentally the same model for attacking the system when the error-free ciphertext is known by the attacker. The one difference is the value of p'. When there are no errors in the ciphertext, $p' = p_1$. With a noisy Q_w, however, p' takes on a different value. It is trivial to show that

$$\begin{aligned} p' &= p_1(1-p_2) + p_2(1-p_1) \\ &= p_1 + p_2 - 2p_1p_2. \end{aligned} \tag{69}$$

Therefore, with only a slight change in parameters, Attacks 1 and 2 can be applied with no change.

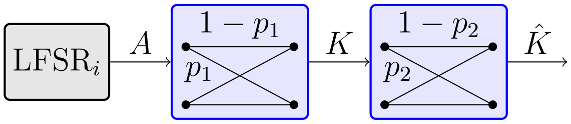

Figure 14: Model relating sequences $\mathbf{A}^n$, $\mathbf{K}^n$, and $\hat{\mathbf{K}}^n$ using a pair of binary symmetric channels.

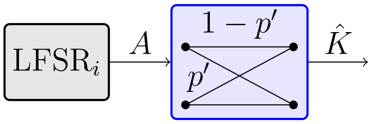

Figure 15: Effective known-plaintext attack model relating the two sequences $\mathbf{A}^n$ and $\hat{\mathbf{K}}^n$.

4.3.1 Mutual Information at the Eavesdropper

Let us briefly analyze the mutual information between a bit in $\mathbf{A}^n$ and its corresponding bit in $\hat{\mathbf{K}}^n$. The entropy of the portion of the secret key allocated to the ith

LFSR is at most ν bits because the order of $g(x)$—and the length of the associating LFSR—is ν; and thus, the entropy of the keystream $\mathbf{K}^n$ is also at most ν bits. Applying Shannon's result shown in (5) in Section 2.3.1 reveals that

$$\mathbb{H}(\mathbf{A}^n) \leq \mathbb{H}(\mathbf{K}^n) \leq \nu. \tag{70}$$

Therefore, an attacker needs only to obtain at most ν bits of information about $\mathbf{A}^n$ to theoretically be able to solve for the secret key. Let us examine how much information about $\mathbf{A}^n$ an attacker can theoretically extract from the error-prone version of the keystream $\hat{\mathbf{K}}^n$. Using the single BSC model with parameter p' we obtain [13, pg.187]

$$\begin{aligned}\mathbb{I}(\mathbf{A}^n; \hat{\mathbf{K}}^n) &= \mathbb{H}(\hat{\mathbf{K}}^n) - \mathbb{H}(\hat{\mathbf{K}}^n | \mathbf{A}^n) \\ &= \mathbb{H}(\hat{\mathbf{K}}^n) - n\mathbb{H}(p') \\ &\leq n\left(1 - \mathbb{H}(p')\right), \end{aligned} \tag{71}$$

where equality holds if $\mathbf{A}^n$ were truly random with equally likely i.i.d. bit realizations, rather than just pseudorandom. The binary entropy function

$$\mathbb{H}(p) = -p \log_2 p - (1-p) \log_2 1 - p \tag{72}$$

takes its maximum value of one when $p = 0.5$ [13]. Thus, not surprisingly, as $p' \to 0.5$ in our system, then $\mathbb{I}(\mathbf{A}^n; \hat{\mathbf{K}}^n) \to 0$, which—as we will show—effectively reduces Attack 1 to a brute-force attack.

4.3.2 Security Enhancement for Attack 1

An estimate for the expected number of trials needed for Attack 1 to succeed was originally derived in [48], and refined in [6]. Suppose Attack 1 is used to attack a system where Eve can only obtain noisy ciphertext and has access to some plaintext. The attack executes normally, as outlined in Section 4.2.1, but with $1 - p'$ as the correlation between $\mathbf{A}^n$ and $\hat{\mathbf{K}}^n$. Recall that the original attack also assumes the setup in Figure 15, but since the attacker had error-free ciphertext in the original scenario, then $p' = p_1$. Now with error-prone ciphertext, noise in the wiretap channel effectively decorrelates the ciphertext with $\mathbf{A}^n$.

The attack chooses the ν bits with the highest values in $(\mathbf{P}^*)^n$ that also form a linearly independent system of equations. The solution to this system of equations is the portion of the secret key allotted to the ith LFSR. If exactly r of the ν bits have been flipped by the BSC, then the maximum number of trials required to cycle through all possible bit patterns up to and including r errors is given by

$$\phi(\nu, r) = \sum_{i=0}^{r} \binom{\nu}{i} \leq 2^{\mathbb{H}(\frac{r}{\nu})\nu}. \tag{73}$$

Let us name the bound as

$$\bar{\phi}(\nu, r) = 2^{\mathbb{H}(\frac{r}{\nu})\nu}. \tag{74}$$

In practice r is not known, but it can be estimated. Let w' be the average number of checks relevant to any one bit, and h' be the maximum integer such that ν bits exist that are expected to satisfy at least h' checks. Then,

$$\bar{r} = \nu - \frac{\nu \sum_{i=h'}^{w'} \binom{w'}{i}(1-p')S^i(1-S)^{w'-i}}{\sum_{i=h'}^{w'} \binom{w'}{i}((1-p')S^i(1-S)^{w'-i} + p'(1-S)^i S^{w'-i})} \tag{75}$$

is an estimate of r, where S is calculated by substituting p' for p_1 in (59). From this equation we learn that $\bar{r}$ of the ν chosen bits are expected to be in error. An estimate on the order of the number of trials required is then given by

$$\phi(\nu, \bar{r}) \leq \bar{\phi}(\nu, \bar{r}) = 2^{\mathbb{H}(\frac{\bar{r}}{\nu})\nu}. \tag{76}$$

In Figure 16, we plot $\bar{\phi}(\nu, \bar{r})$ for several p_2 values over the range of p_1, using (69) to calculate p' as a function of p_1 and p_2. The results in Figure 16 are for an LFSR with length $\nu = 32$, $t = 6$ feedback connections, and $n = \nu \times 10^6$ bits of known plaintext.

Simulated attacks provide some idea as to the accuracy of $\bar{\phi}(\nu, \bar{r})$. These are shown in Figure 17. The expression $\bar{\phi}(\nu, \bar{r})$ approximates the simulated result when p' is close to zero. When $p' = 0.5$, however, then $\mathbb{H}(\bar{r}/\nu) = 1$ and $\bar{\phi}(\nu, \bar{r}) = 2^{\mathbb{H}(\bar{r}/\nu)\nu} = 2^{\nu}$. The

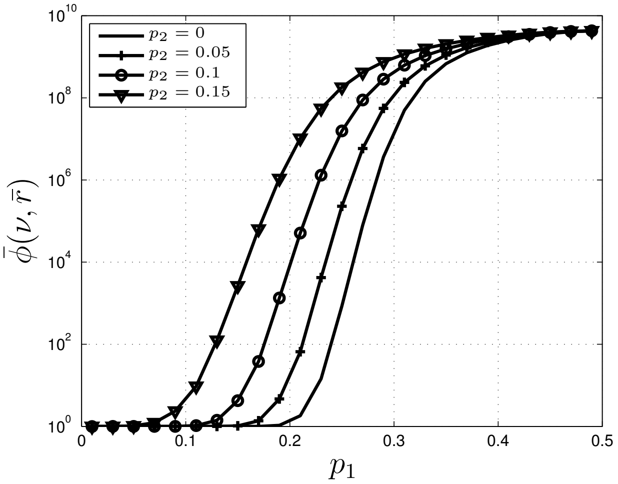

Figure 16: Expected bound $\bar{\phi}(\nu, \bar{r})$ on the number of trials required to find the secret key using Attack 1 when $\nu = 32$, $n = \nu \times 10^6$, and $t = 6$.

simulated average at $p' = 0.5$ is $2^{\nu-1}$, one half of $\bar{\phi}(\nu, \bar{r})$ when $p' = 0.5$, which is also the average of a straight forward brute-force attack. This result indicates that noise in the ciphertext has the effect of decorrelating $\mathbf{A}^n$ and $\hat{\mathbf{K}}^n$. We also see shifts in the plot for different values of p_2. Clearly, p' increases with either p_1 or p_2, and these results indicate that as p' increases, then the expected work for Attack 1 to succeed must increase as well.

Let us now define the per-letter mutual information between $\mathbf{A}^n$ and $\hat{\mathbf{K}}^n$ as

$$I_{A\hat{K}} = \frac{1}{n}\mathbb{I}(\mathbf{A}^n, \hat{\mathbf{K}}^n), \tag{77}$$

and realize that the bound in (71) can be applied to this quantity as well so that

$$I_{A\hat{K}} \leq 1 - \mathbb{H}(p') = \bar{I}_{A\hat{K}}. \tag{78}$$

We can now plot our results as a function of this upper bound on $I_{A\hat{K}}$ to give us some idea as to how mutual information scales the effectiveness of cryptographic attacks. The trends are quite clear, and perhaps expected. As $p' \to 0.5$, then the upper bounds on both $\mathbb{I}(\mathbf{A}^n; \hat{\mathbf{K}}^n)$ and $I_{A\hat{K}}$ go to zero, forcing the mutual information to zero along with the bounds. When this occurs, the number of required iterations to recover the

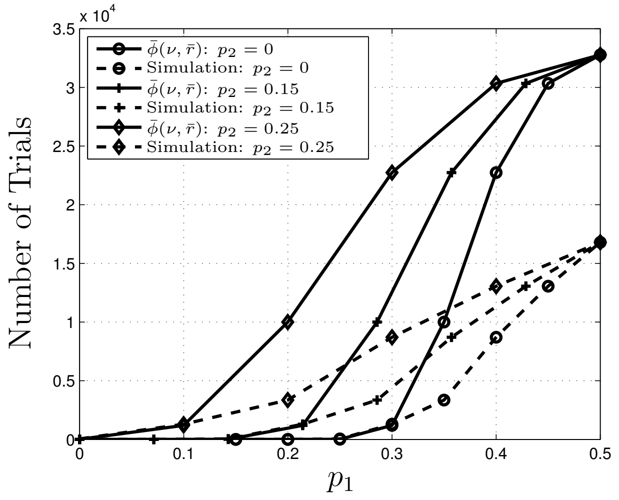

Figure 17: Results from simulations of Attack 1 showing necessary computations to crack the LFSR-based cryptographic system. Here $\nu = 15$, $N' = 1500$, and $t = 4$.

secret key using Attack 1 goes to $2^{\nu-1}$, which signifies brute-force complexity. Figure 18 shows the expected number of trials as a function of the bound on per-letter mutual information when $\nu = 15$, $t = 4$, and $n = 1500$.[4]

4.3.3 Security Enhancement for Attack 2—EXIT Charts

In the case of Attack 1, the bit realizations of the random vector $\hat{\mathbf{K}}^n$ remain constant throughout the attack. The information about $\mathbf{A}^n$ imbedded in $\hat{\mathbf{K}}^n$ is extracted and combined with knowledge of the structure of $\mathbf{A}^n$ to find the secret key. However, in Attack 2 the values of bits in $\hat{\mathbf{K}}^n$ are modified at the end of each round, thus altering the density on $\hat{\mathbf{K}}^n$ as the attack progresses. Let $(\hat{\mathbf{K}}^n)^{[l]}$ be the $\hat{\mathbf{K}}^n$ sequence after the bit flipping in round l of Attack 2. Say the attack takes J rounds for $(\hat{\mathbf{K}}^n)^{[l]}$ to either stagnate or converge to $\mathbf{A}^n$. Then an information-theoretic analysis of Attack 2 requires knowledge of $\mathbb{I}(\mathbf{A}^n; (\hat{\mathbf{K}}^n)^{[l]})$ for $l = 0, 1, \ldots, J$. Since sequences are binary, we expect the per-letter mutual information between $\mathbf{A}^n$ and $(\hat{\mathbf{K}}^n)^{[l]}$ to go to one as l goes to J in a successful attack, and to converge to a value less than one when an

[4] Here t is small relative to ν for ease in simulation, but these trends extend to larger t.

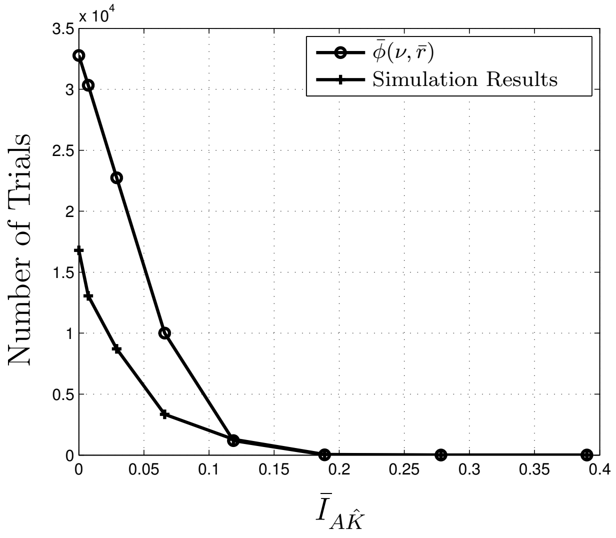

Figure 18: Number of trials required for a successful attack versus $\bar{I}_{A\hat{K}}$, an upper bound on the per-letter mutual information between $\mathbf{A}^n$ and $\hat{\mathbf{K}}^n$. The order of $g(x)$ is $\nu = 15$, the number of nonzero coefficients in $g(x)$ is $t = 4$, and the amount of known plaintext bits $n = 1500$.

attack fails.

Actual calculations of $\mathbb{I}(\mathbf{A}^n; (\hat{\mathbf{K}}^n)^{[l]})$ prove to be difficult, therefore, we will approximate these values by assuming that bits in $\mathbf{A}^n$ are i.i.d and uniformly distributed over $\{0,1\}$. We call the mutual information calculated under this assumption $\tilde{\mathbb{I}}(\mathbf{A}^n; (\hat{\mathbf{K}}^n)^{[l]})$, and define the per-letter mutual information under this assumption as

$$\tilde{I}_{A\hat{K}}^{[l]} = \frac{1}{n}\tilde{\mathbb{I}}(\mathbf{A}^n; (\hat{\mathbf{K}}^n)^{[l]}). \tag{79}$$

We will use EXIT charts to show the expected progression of $\tilde{I}_{A\hat{K}}^{[l]}$ as l ranges from zero up to J. EXIT analysis provides intuition on the decoding threshold in terms of BER in the ciphertext by noting the lowest error rate that first introduces a cross in the plotted intrinsic versus extrinsic information curves. The *intrinsic* information can be defined as the information available at the input of a decoding iteration. The *extrinsic* information is then defined as the information available at the output of a decoding iteration. Prior to building EXIT charts, however, we should note that another technique for anticipating success in Attack 2 was given in [48], by determining the expected result of the first iteration of the algorithm. It was observed that if the first iteration obtains additional information about the keystream, then eventually the iterative attack converges on the correct data sequence. In other words, the first step's outcome seems to be sufficient to estimate the algorithm's result. We now calculate the threshold p_{thr} to maximize the probability that $\hat{K}_j \neq A_j$ given that $P_j^* < p_{thr}$. Let N_w be the expected number of bits such that both $\hat{K}_j \neq A_j$ and $P_j^* < p_{thr}$, and let N_v be the expected number of bits such that $\hat{K}_j = A_j$ and $P_j^* < p_{thr}$, for $j = 1, 2, \ldots, n$. Also, let $N_i = N_w - N_v$. If N_{c_0} represents the total number of bits such that $\hat{K}_j = A_j$ prior to any iterations, then the toggling of all bits with $P_j^* < p_{thr}$ will result in an expected $(N_{c_0} + N_i)$ correct bits. Obviously if N_i is negative, then the expected outcome of the first iteration will leave more bits in error than were originally so, and therefore, according to [48] will cause the attack to fail.

To ensure that the algorithm does not eventually converge on the correct sequence, it must be guaranteed that Attack 2 has no correction capability. Strictly speaking, this is a difficult guarantee; however, we will adopt the nomenclature of [48] and say that Attack 2 has *correction capability zero* if $N_i \leq 0$. The correction ratio

$$F = \frac{N_i}{N_w + N_v} = \frac{N_w - N_v}{N_w + N_v} \tag{80}$$

is used to scale the value of N_i to a real number within the range of $[-1, 1]$ while maintaining its sign. Figure 19 shows the value of F for several BSC parameters p_2, over a range of p_1 values. Simulations of Attack 2 give some evidence that $F \leq 0$ is sufficient to predict attack failure; however, results also show that $F > 0$ is not sufficient to guarantee attack success.

Example 2. Let the primitive connection polynomial for the ith LFSR be $g(x) = 1 + x + x^2 + x^3 + x^{12} + x^{21} + x^{31}$, and $p_1 = \Pr(A_j \neq K_j) = 0.2$. In the first of two cases, $p_2 = 0$, i.e., the error rate in the wiretap channel is zero. Therefore, $p' = p_1 = 0.2$ and F is calculated using (80) to be 0.826. Case two sets $p_2 = 0.1$, meaning the

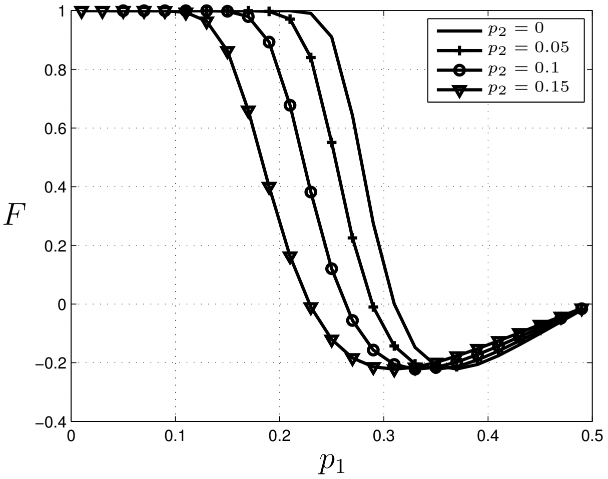

Figure 19: Correction ratio F of Attack 2 for $\nu = 32$, $N' = \nu \times 10^6$, and $t = 6$. $F \leq 0$ indicates that an attack will likely fail.

BER in $\hat{\mathbf{K}}^n$ is 0.1. Using (69) we calculate $p' = 0.26$, and using (80) we find that $F = -0.034$. These values of F imply that Attack 2 will succeed in case one and fail in case two. Actual outcomes of these attacks are shown in Table 1. Case one does indeed converge to $\mathbf{A}^n$ in 16 rounds, while case two requires 34 rounds before the algorithm stagnates and fails. Note that in the failed case, most rounds result in fewer correct bits than the previous round.

4.4 EXIT Charts Based on Hard Decisions

EXIT charts provide a closer look into the information transfer that occurs during an iterative decoding process, and therefore, can be applied to Attack 2. The first of two methods for generating EXIT charts for Attack 2 tracks per-letter mutual information between $\mathbf{A}^n$ and $(\hat{\mathbf{K}}^n)^{[l]}$ from round to round by assuming bits in $\mathbf{A}^n$ to be i.i.d. and uniformly distributed over $\{0, 1\}$. This is not such a bad assumption for this attack, since the algorithm recovers the entire sequence $\mathbf{A}^n$ anyway. When we calculate mutual information under this assumption, we use the notation set forth in (79). To be clear, we consider elements in $\mathbf{A}^n$ to be realizations of a single random

Table 1: Simulation results of Attack 2 comparing scenarios with and without added security from the physical layer. For these simulations, $\nu = 31$, $n = \nu \times 100$, $t = 6$, and $p_1 = 0.2$.

	Case 1: $p_2 = 0$		Case 2: $p_2 = 0.1$	
Round Index	Number of bits flipped	Total correct bits	Number of bits flipped	Total correct bits
1	30	2487	1	2276
2	91	2526	3	2277
3	122	2586	6	2277
4	42	2628	8	2275
5	50	2676	11	2268
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
14	43	3075	2	2204
15	23	3098	100	2164
16	2	3100	4	2164
$\vdots$	-	-	$\vdots$	$\vdots$
34	-	-	1	2079
35	-	-	0	2079
$\vdots$	-	-	0	2079

variable A, and elements in $(\hat{\mathbf{K}}^n)^{[l]}$ to be realizations of a single random variable $\hat{K}^{[l]}$. Then we calculate the per-letter mutual information $\tilde{I}^{[l]}_{A\hat{K}}$ from (79) using the expression

$$\tilde{I}^{[l]}_{A\hat{K}} = \mathbb{I}(A; \hat{K}^{[l]}) = \sum_{a\in\{0,1\}} \sum_{\hat{k}\in\{0,1\}} p_{A\hat{K}^{[l]}}(A = a, \hat{K}^{[l]} = \hat{k}) \log_2 \frac{p_{A\hat{K}^{[l]}}(A = a, \hat{K}^{[l]} = \hat{k})}{p_A(A = a) p_{\hat{K}^{[l]}}(\hat{K}^{[l]} = \hat{k})} \tag{81}$$

for rounds from $l = 0, 1, \ldots, J$.

The calculation of $\tilde{I}^{[l]}_{A\hat{K}}$ requires us to estimate the probability mass function of $\hat{K}^{[l]}$, as well as the joint mass function of $\mathbf{A}^n$ and $\hat{K}^{[l]}$. Since the channel we are considering is symmetric, all of this can be done through simulation by counting bits that are still in error at the end of each round and dividing by the total number of bits. Realize that the increase in information due to round l is $(\tilde{I}^{[l]}_{A\hat{K}} - \tilde{I}^{[l-1]}_{A\hat{K}})$; therefore, we assign the intrinsic and extrinsic information for round l as $\tilde{I}^{[l-1]}_{A\hat{K}}$ and $\tilde{I}^{[l]}_{A\hat{K}}$, respectively. The EXIT chart portrays the expected increase in information by plotting $\tilde{I}^{[l-1]}_{A\hat{K}}$ versus $\tilde{I}^{[l]}_{A\hat{K}}$ for curve one, while the second curve in the EXIT chart is $\tilde{I}^{[l]}_{A\hat{K}}$ versus $\tilde{I}^{[l-1]}_{A\hat{K}}$. Thus the expected progress of the decoder is shown by reflecting back and forth between curves. If $\tilde{I}^{[l]}_{A\hat{K}}$ goes to one, then the attack converges on the correct sequence; therefore, there must exist a gap between the two curves if a successful attack is to be expected on average.

To show average tendencies in the mutual information during Attack 2 by simulation, we construct EXIT charts using a binning technique. The value $\tilde{I}^{[l]}_{A\hat{K}}$ is calculated for every round in a large number of attacks. Then the expected increase in information is obtained for each section of the chart by subdividing the x-axis into Δ equal segments or bins. The data are sorted according to intrinsic information, and then the extrinsic information is averaged in each bin. The center of each of the Δ segments is used as the intrinsic information for the corresponding bin when forming the chart.

Results using this method for a particular set of system parameters averaged over 100 attacks are shown in Figure 20. For this example, we assume that $p_2 = 0$, and that the correlation in the keystream generator is such that $p_1 = 0.2$. We observe that the EXIT chart predicts an overall tendency for the attack to succeed, which is implied by the gap between curves. We also note that the EXIT curves do not extend to zero. Generating these curves was implemented through simulating attacks on the system. Although some rounds did yield a negative correction, none resulted in zero extrinsic information; therefore, no rounds exhibited zero intrinsic information either, leaving bins around zero empty. Finally we observe that the gap between EXIT curves is narrower for lower intrinsic information regimes. This fact defends the technique used in [48] and [6] in defining the *correction capability* of Attack 2 using only the expected results in the first round of an attack. If the first round provides good correction, then the chart indicates that convergence to $\mathbf{A}^n$ is likely to proceed quickly. When the first round exhibits mediocre or poor correction, the algorithm must proceed through the pinched region of the gap resulting in slower convergence on average.

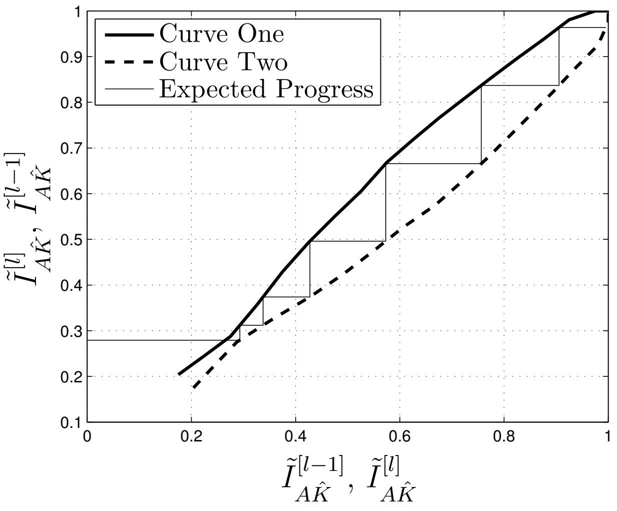

Figure 20: EXIT chart with $\Delta = 20$ formed by averaging the hard-decision output of 100 simulations of Attack 2 with $\nu = 31$, $t = 6$, $\alpha = 5$, $n = 3100$, and $p_1 = p' = 0.2$.

Another EXIT chart for a similar setup as that in Figure 20 is shown in Figure 21, the only difference being that $p_2 = 0.1$ in the latter example, while $p_2 = 0$ in the former example. These two figures are the EXIT charts for the same scenarios as in Example 2. Figure 21 shows that Attack 2 is likely to fail when $p_2 = 0.1$ because of the crossover in the EXIT chart. Again this behavior can be predicted from the average correction in the first round. Figure 21 portrays more errors on average following the first round than there were prior to launching the attack. In this scenario, the expected progress of an attack converges on the crossover point in the EXIT chart rather than converging to one as in Figure 20. The tabulated results from Example 2 showed a similar result. From Figures 20 and 21, we can deduce that a physical-layer security code that ensures a 10% bit error rate in Eve's ciphertext is sufficient on average to prevent Attack 2 from obtaining the secret key.

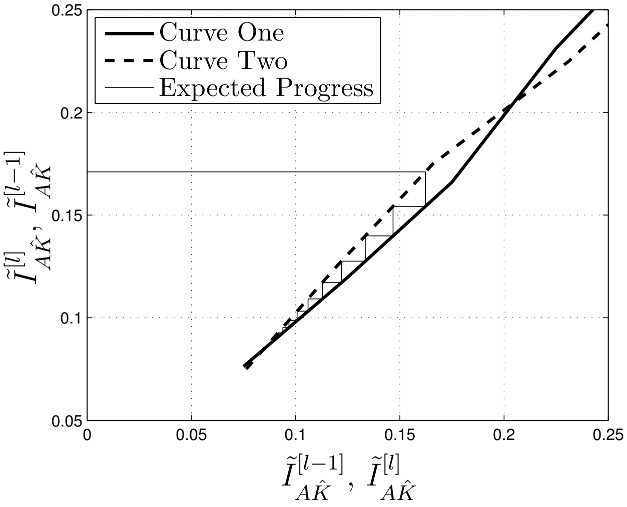

Figure 21: EXIT chart with $\Delta = 20$ formed by averaging the hard-decision output of 100 simulations of Attack 2 with $\nu = 31$, $t = 6$, $\alpha = 5$, $n = 3100$, $p_1 = 0.2$, and $p_2 = 0.1$ yielding $p' = 0.26$.

4.5 EXIT Charts Based on Densities

The second technique for generating EXIT charts requires the use of message-passing parameters and estimating densities of continuous random variables [23, 12]. If we consider the attack in terms of its underlying bipartite graph G, we let check nodes comprise the vertices of one bipartition, and bit nodes comprise the vertices of the other bipartition. A check node is adjacent to a bit node if and only if the check expression includes that particular bit. In practice only values from $(\mathbf{P}^*)^n$ and $\mathbf{S}^{(t+1)\times w'}$ are passed between bit nodes and check nodes at each iteration, but to track the mutual information at each iteration, we must form useful probability measures. We adopt the notion of using likelihood differentials similar to those used in LDPC message passing [23]. Suppose the algorithm requires L total iterations to converge, i.e., L is the sum of the iterations in all J rounds required for convergence. Let $\mathbf{S}^{[l]}$ and $(\mathbf{P}^*)^{[l]}$ denote the values of $\mathbf{S}^{(t+1)\times w'}$ and $(\mathbf{P}^*)^n$, respectively, after the update in the lth iteration, where the superscripts indicating size are omitted from the new notation for practicality. Consider the uth bit of the vth check, where the vth check is given by (54). Then the messages being passed from check nodes to bit nodes along the edges of G are calculated using (64), and are given as

$$\begin{aligned}\delta S_{u,v}^{[l]} &= \Pr\left(A_{j+j_u} = 0|(\mathbf{P}^*)^{[l-1]}\right) - \Pr\left(A_{j+j_u} = 1|(\mathbf{P}^*)^{[l-1]}\right)\\ &= \begin{cases}(1 - S_{u,v}^{[l]}) - S_{u,v}^{[l]} & \text{if } V = 1\\ S_{u,v}^{[l]} - (1 - S_{u,v}^{[l]}) & \text{if } V = 0\end{cases}\end{aligned} \tag{82}$$

where

$$V = \sum_{x \in \{j_0, \ldots, j_{u-1}, j_{u+1}, \ldots, j_t\}} \hat{K}_{j+x}. \tag{83}$$

Note that $\mathbf{S}^{[0]}$ is calculated using (59) where p' is substituted for p_1. The algorithm then calculates $(\mathbf{P}^*)^{[0]}$ using (60) and the information in $\mathbf{S}^{[0]}$, and passes back along the edges of the bipartite graph new bit to check information messages

$$\begin{aligned}\delta(P^*)_j^{[l]} &= \Pr\left(A_j = 0|\mathbf{S}^{[l]}\right) - \Pr\left(A_j = 1|\mathbf{S}^{[l]}\right)\\ &= \begin{cases}(1 - (P_j^*)^{[l]}) - (P_j^*)^{[l]} & \text{if } \hat{K}_j = 1\\ (P_j^*)^{[l]} - (1 - (P_j^*)^{[l]}) & \text{if } \hat{K}_j = 0\end{cases}\end{aligned} \tag{84}$$

for $j = 1, 2, \ldots, n$, and for $l = 0, 1, \ldots, L$.

To form the EXIT chart, we once again calculate per-letter mutual information as discussed in Section 4.3.3, and adhere to the notation set forth in (79). Therefore, A is modeled as a random variable with i.i.d. uniformly distributed realizations on $\{0, 1\}$, and the entries of $\mathbf{A}^n$ are assumed to be realizations of A. Also, $\delta S^{[l]}$ is a random variable that governs the continuous distribution on values in $\delta\mathbf{S}^{[l]}$. The per-letter mutual information between A and $\delta S^{[l]}$ will be denoted as

$$\tilde{I}_{A,\delta S}^{[l]} = \mathbb{I}(A; \delta S^{[l]}). \tag{85}$$

Likewise, let $\delta(P^*)^{[l]}$ be a random variable that governs the distribution over all entries in $\delta(\mathbf{P}^*)^{[l]}$. We will denote per-letter mutual information between A and $\delta(P^*)^{[l]}$ as

$$\tilde{I}_{A,\delta P^*}^{[l]} = \mathbb{I}(A; \delta(P^*)^{[l]}) \tag{86}$$

for $l = 0, 1, \ldots, L$. Therefore, we must solve for mutual information between a discrete binary random variable A, and continuous random variables $\delta S^{[l]}$ and $\delta(P^*)^{[l]}$. From [12], if A is discrete over the alphabet $\mathcal{A}$ and B is a continuous random variable over the region $\mathcal{B}$ then

$$\begin{aligned}\mathbb{I}(A;B) &= \int_{b\in\mathcal{B}} \sum_{a\in\mathcal{A}} p_{AB}(a,b)\log_2 \frac{p_{AB}(a,b)}{p_A(a)p_B(b)} db \\ &= \sum_{a\in\mathcal{A}} \int_{b\in\mathcal{B}} p_{B|A}(b|a)p_A(a)\log_2 \frac{p_{B|A}(b|a)}{\sum_{a'\in\mathcal{A}} p_{B|A}(b|a')p_A(a')} db. \qquad (87)\end{aligned}$$

If A is binary with equally likely prior probabilities, then this becomes

$$\begin{aligned}\mathbb{I}(A;B) =& \frac{1}{2}\int_{b\in\mathcal{B}} \big(p_{B|A}(b|a=0) + p_{B|A}(b|a=1)\big) \times \qquad (88) \\ & \times \big(1 - \log_2(p_{B|A}(b|a=0) + p_{B|A}(b|a=1))\big) + \\ & + \sum_{j\in\{0,1\}} p_{B|A}(b|a=j)\log_2(p_{B|A}(b|a=j))db. \qquad (89)\end{aligned}$$

Thus, $\tilde{I}^{[l]}_{A,\delta S}$ and $\tilde{I}^{[l]}_{A,\delta P^*}$ can be calculated. If we assumed further that the distributions of $\delta S^{[l]}$ and $\delta(P^*)^{[l]}$ were Gaussian, then we could solve for a closed-form expression; however, histograms of these data appear to be Gaussian only in the first iteration of each round in the fast-correlation attack. Therefore, the mutual information calculations are performed numerically using histograms as estimates of density functions.

This type of EXIT chart takes into account the individual iterations in each round, rather than just the hard decisions following the rounds. Bit adjustments introduce abrupt changes in the mutual information, and thus affect the shape of the EXIT chart. To provide smooth curves, a length-two finite impulse response averaging filter is convolved with the data and the binning technique mentioned in Section **??** is implemented on the filtered data.

To form the EXIT chart based on density estimation, we plot the first curve as $\tilde{I}^{[l]}_{A,\delta S}$ versus $\tilde{I}^{[l]}_{A,\delta P^*}$ for all iterations. The second curve is $\tilde{I}^{[l+1]}_{A,\delta S}$ versus $\tilde{I}^{[l]}_{A,\delta P^*}$, thus vertical lines in the expected path through the chart represent the information transfer between check nodes and bit nodes in the same iteration, and horizontal lines represent the information transfer between the bit nodes of the current iteration and the check nodes of the next iteration [12].

Examples of EXIT charts generated using this second method are now presented using the same LFSR as before, with connection polynomial $g(x) = 1 + x + x^2 + x^3 + x^{12} + x^{21} + x^{31}$. Again, $\alpha = 5$, $n = 3100$, and $t = 6$. We also set $p_1 = 0.2$ implying that $\Pr(a_j = k_j) = 0.8$. Let us consider again the case where $p_2 = 0$ yielding $p' = 0.2$, and compare it to the case where $p_2 = 0.05$ yielding $p' = 0.23$. Plots are given in Figures 22 and 23, respectively. Since this method takes into account each iteration, it therefore requires more steps to traverse the expected path in Figure 22. The crossover point in Figure 23 indicates that the attack will have an inclination to stagnate, and thus fail to obtain the sequence $\mathbf{A}^n$ when $p' = 0.23$. In these examples a mere five percent error rate is required to alter the expected behavior of the attack.

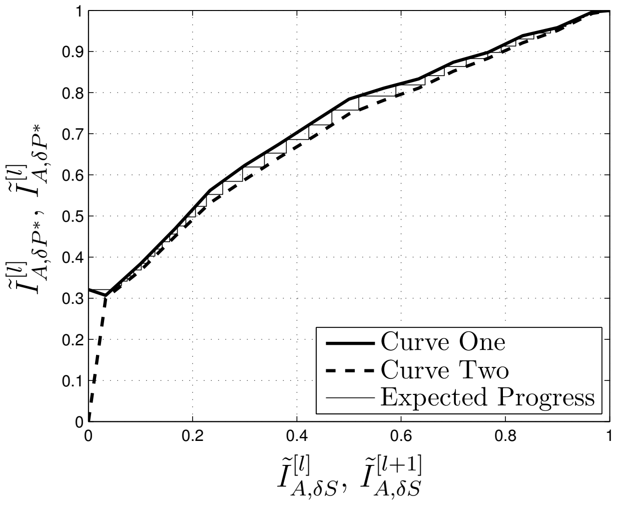

Figure 22: EXIT chart with $\Delta = 15$ formed by averaging soft-decision output of 100 simulations of Attack 2 with $\nu = 31$, $t = 6$, $\alpha = 5$, $n = 3100$, $p_1 = 0.2$, and $p_2 = 0$, yielding $p' = 0.2$.

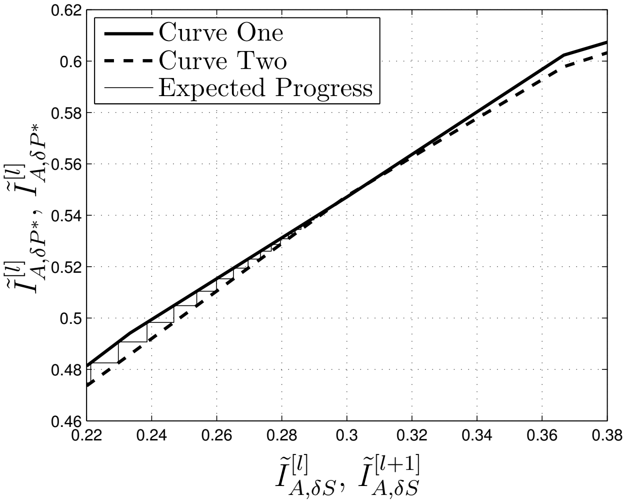

Figure 23: EXIT chart with $\Delta = 15$ formed by averaging soft-decision output of 100 simulations of Attack 2 with $\nu = 31$, $t = 6$, $\alpha = 5$, $n = 3100$, $p_1 = 0.2$, and $p_2 = 0.05$ yielding $p' = 0.23$. The viewable range is adjusted to show the crossover point.

4.6 Discussion and Conclusions

This chapter has analyzed the effectiveness of known attacks against LFSR-based stream ciphers when an eavesdropper has access to error-prone ciphertext and known plaintext. The wiretap channel was modeled as a BSC with a certain probability of bit errors occurring randomly in the channel. It was observed that in known plaintext attacks, noisy ciphertext directly implies noisy keystream data. Solving for a specific output sequence of one of the LFSRs that makes up the keystream generator is then made more difficult because of the uncertainty in the keystream.

We showed that this analysis changed only a single parameter in fast-correlation attacks against this system, because the attack assumes an effective BSC separating a specific LFSR output sequence in the keystream generator and the true keystream. That assumption of correlation allowed us to incorporate additional confusion from error rates in the wiretap channel.

The end results were expressions for understanding the necessary error rates in ciphertext to drive Attack 1 to a brute-force attack, and causing the iterative algorithm in Attack 2 to fail completely. We analyzed parameters of these attacks that indicate their effectiveness when ciphertext was error prone. We also applied the tool of EXIT charts to verify the results for Attack 2. For this attack, a small bit-error rate, say 5% was shown to effectively shut down the convergence of the attack in some cases. It should be pointed out that the physical-layer security codes we will present in Chapter 5 will far exceed these error rates in the eavesdropper's ciphertext for almost all configurations of channel parameters when an eavesdropper may experience packet losses in data transmission. The findings of this chapter imply that if physical-layer security codes can be used against eavesdroppers to impart even small error rates in the eavesdropper's obtained ciphertext, then stream ciphers can enjoy a security enhancement from the physical layer.

CHAPTER V

STOPPING SET CODES FOR PHYSICAL-LAYER SECURITY

Physical-layer security coding has the potential to enhance cryptographic protocols by hiding the ciphertext from potential attackers. However, the problem of combined physical-layer security with cryptography requires two problems to be solved. The first problem of cryptanalysis of error-prone ciphertext was addressed in Chapters 3 and 4. This chapter addresses the second problem of developing practical codes that offer physical-layer security. As was seen in Chapter 4, some ciphers can be made much stronger if we can design codes that guarantee even a small error rate in an eavesdropper's observed ciphertext. The codes presented here are based on low-density parity-check codes, and exploit a point of failure in message passing decoding called stopping sets, for security. Much of the material in this chapter has been published in [9, 10, 11].

5.1 Motivation and Outline

This chapter provides a new encoder design that looks at offering legitimate parties security, even if system parameters are unknown during the design phase of the codes. The encoder aims to provide a blanket of security that covers almost every conceivable set of channel parameters over the packet erasure wiretap channel model. The security analysis calculates the probability density function on D, a random variable that represents the number of degrees of freedom that exist in an eavesdropper's information about the ciphertext. Degrees of freedom were previously introduced in Section 2.5.2.3. There it was pointed out that computational security and information-theoretic security can be addressed using D. In this chapter, computational security is shown to grow as $\mathbb{E}[\frac{1}{2}(2^D + 1)]$ for our binary codes. We will also show that $\mathbb{E}[D]$ is equal to the equivocation for the prescribed encoder.

Our encoder alleviates the shortcomings of many practical physical-layer security codes, as discussed in Section 2.5.2.2. The design is robust to varying channel parameters, imperfect channel state information (CSI) at the encoder, and non-degraded system models. In fact, the family of codes that we have designed even offer security enhancement to cryptography when attackers have an advantage in signal quality over legitimate receivers. The scheme initially assumes erasure events over Q_m and Q_w are statistically independent, and relies on a nonsystematic LDPC code design, with puncturing and interleaving steps in the encoder. Legitimate receivers employ automatic repeat-request (ARQ) for reliability through an authenticated public feedback channel. Finally, the scheme requires no secret key and no rate reduction in data transmission.

The system model is given in Section 5.2, while the background for low-density parity-check codes and stopping sets is discussed in Section 5.3. End-to-end details of the encoder (Section 5.4) and decoder (Section 5.5) are provided. Design criteria are also specified to maximize the degrees of freedom in the maximum-likelihood attack as well as the message-passing attack. The security analysis is then given in Section 5.6, which includes bounds on the increase in computational secrecy of an underlying cryptosystem, as well as extensions to cover broadcast scenarios with multiple receivers and multiple collaborative attackers. We examine the case where packet erasures are correlated across Q_m and Q_w in Section 5.7. Ultimately, we investigate the error rates in the ciphertext when the encoder provides security in Section 5.8, and include some discussion of the material in the chapter in Section 5.9.

5.2 System Model and Degrees of Freedom

We begin by reducing the combined coding and cryptographic system shown in Figure 5 to only the essential pieces for this problem. The wiretap channel model [4] with feedback is shown in Figure 24. In this model, the input message is binary, compressed, and encrypted and is labeled E in the figure. In fact, all signals will be binary in this chapter. It will be helpful to think of the encrypted message as being broken up into L blocks of length k, where k is called the *dimension* of the encoder, and indicates that the block encoder presented in this chapter encodes k bits at a time. Let us continue our trends in notation, and define

$$\mathbf{E}^{kL} = (\mathbf{E}_1^k, \mathbf{E}_2^k, \ldots, \mathbf{E}_L^k), \tag{90}$$

where

$$\mathbf{E}_i^k = (E_{(i-1)k+1}, E_{(i-1)k+2}, \ldots, E_{ik}). \tag{91}$$

To avoid some of the cumbersome aspects of this notation, we will sometimes simply write $\mathbf{E}$ to indicate the entire input message. Other signals in the system model follow the same guidelines. As a guide to this notation, recall from Section 2.1 that the superscripts on bold capital letters signify the length of the random vector, while subscripts indicate indices. Since the bits of the message are modeled as random, we denote them with capital letters as well. A realization of a random vector $\mathbf{E}_i^k$ must come from $\mathcal{E}^k$, the alphabet of all binary length-k vectors. Let n be the blocklength of the encoder. Then the coding rate is k/n. Since $\mathbf{E}$ was compressed prior to encryption, all possible k-length bit combinations from $\mathcal{E}^k$ are equally likely for the blocks $\mathbf{E}_i^k$ when $i = 1, 2, \ldots, L$.

The encoder design will be discussed at length in Section 5.4, but as a precursor, the encoder outputs packets. Each packet possesses α bits from L different input blocks of data. Thus, the size of each packet is αL. Define η to be the number of transmitted packets so that $\mathbf{X}^{\eta\alpha L} = (\mathbf{X}_1^{\alpha L}, \mathbf{X}_2^{\alpha L}, \ldots, \mathbf{X}_\eta^{\alpha L})$. We continue to call the

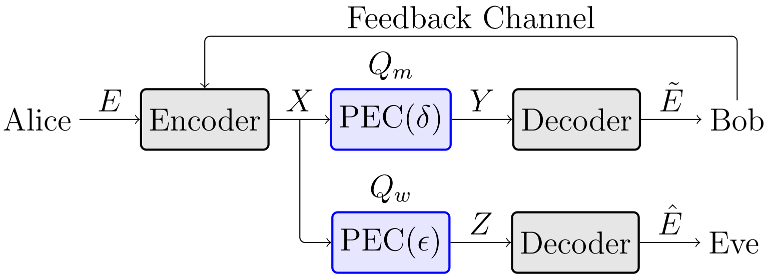

Figure 24: Wiretap channel model with feedback assuming packet erasure channels for both the main channel Q_m and the wiretap channel Q_w.

main channel Q_m and the wiretap channel Q_w. The two channels are packet erasure channels (PECs) throughout this chapter, where packets are erased randomly with probability δ in Q_m, and with probability ϵ in Q_w. The output collection of packets from Q_m is called $\mathbf{Y}$ and the output collection of packets from Q_w is called $\mathbf{Z}$. Finally, the decoder takes the received packets and generates an estimate on $\mathbf{E}$, which we call $\tilde{\mathbf{E}}$ for Bob, and $\hat{\mathbf{E}}$ for Eve.

Notice that there are no restrictions on ϵ. In fact, Q_w may be of higher quality than Q_m in this model. However, the feedback channel will more than compensate for most scenarios when Eve has an advantage of channel quality. The encoder and decoder exploit the independent nature of erased packets across Q_m and Q_w (although it will be shown that the design can still be effective with correlated erasures). Of course, the system must guarantee that $\tilde{\mathbf{E}} = \mathbf{E}$, while at the same time making Eve as ignorant as possible. The authenticated feedback channel available to Bob plays a key role in accomplishing both of these endeavors. This public noiseless channel is used to request the retransmission of erased packets. Since it is authenticated, Alice is able to deduce whether Bob sent the request, and can detect any tampering with the data [14], thus restricting Eve to passive status [57]. Requests by Bob are public, and there is no secret key employed at the physical layer. The sole source of confusion for Eve is her own naturally occurring erasure pattern across Q_w.

Cryptographic attacks often assume an attacker has the luxury of an error-free version of $\mathbf{E}$ (or even some of the plaintext), but our design aims to prevent this from occurring, by creating degrees of freedom in the attacker's knowledge of $\mathbf{E}$ by means of physical-layer security.

Definition 11. The number of *degrees of freedom* in a received codeword is a random variable D that takes on the number of encoded symbols for which an eavesdropper has no information. Therefore, the probabilities of all symbol values on these D symbols are equally likely.

For binary codes with $D = d$, the decoder will determine 2^d equally likely codewords of length n, each mapping to a unique k-bit encrypted message in $\mathcal{E}^k$. Since we assume that the attacker knows the encoder, the maximum value of D is k because a block in $\mathbf{E}$ has k information bits associated with it and an eavesdropper can, at least in theory, reproduce the codeword with k bits of information. We equated k degrees of freedom in a block encoder such as this to the notion of *perfect secrecy* in Section 2.5.2.3. Therefore, degrees of freedom are similar to the information-theoretic secrecy metric of general equivocation, which we will define in this chapter to be $\mathbb{H}(\mathbf{X}|\mathbf{Z})$. In fact, if we have D degrees of freedom in k bits, and the other $k - D$ bits are known perfectly, then the entropy of the block is exactly D bits. Since an attacker has no knowledge of the bits associated with the D degrees of freedom, the average number of guesses required to obtain them is equal to the mean of a discrete uniform random variable that ranges over 2^D values. Since D itself is a random variable, then

$$\mathbb{E}[\frac{1}{2}(2^D + 1)] = \frac{1}{2}(\mathbb{E}[2^D] + 1) \tag{92}$$

guesses must be made on average to resolve D degrees of freedom. Using this reasoning, the goals of our physical-layer design are as follows: first, to ensure that $D = 0$ for Bob so that $\tilde{\mathbf{E}} = \mathbf{E}$; second, to make D as large as possible for Eve; and third, to ensure that attacks on the cryptogram fail if the Hamming distance

$$d_H(\hat{\mathbf{E}}, \mathbf{E}) > 0. \tag{93}$$

5.3 LDPC Codes and Stopping Sets

We employ low-density parity-check (LDPC) codes [23], and exploit the phenomenon of stopping sets in our encoder design. This section introduces the codes and their properties.

5.3.1 LDPC Codes

Let us define a general binary LDPC code $\mathcal{C}$ with blocklength n', and dimension k. Note that this k is identical to k from section 5.2, but n' the blocklength of the LDPC code, is different from n the blocklength of the encoder. Later, we will discuss the role of puncturing in our encoder to reduce the effective blocklength from n' to n. The parity check matrix $\mathbf{H}^{(n'-k)\times n'}$ fully defines the code. We will find it helpful to think of $\mathbf{H}^{(n'-k)\times n'}$ in terms of its corresponding Tanner graph $G_{\mathcal{C}}$ [12, 58]. The set of variable nodes is

$$V = (v_1, v_2, \ldots, v_{n'}), \tag{94}$$

while the set of check nodes is

$$U = (u_1, u_2, \ldots, u_{n'-k}). \tag{95}$$

Variable nodes correspond to the n' bits in a codeword, and to columns in $\mathbf{H}^{(n'-k)\times n'}$. Checks correspond to rows in $\mathbf{H}^{(n'-k)\times n'}$, where the set of bits that participate in the check u_i is denoted [12]

$$\mathcal{N}_i = \{j : H_{i,j} = 1\}. \tag{96}$$

Then the ith check is calculated in $GF(2)$ as

$$u_i = \sum_{j \in \mathcal{N}_i} v_j = 0. \tag{97}$$

The notation $\mathcal{N}_{i,j}$ signifies all bits in the ith check except the jth bit. The jth variable node shares an edge with the ith check node in $G_{\mathcal{C}}$ if and only if $j \in \mathcal{N}_i$.

Example 3. The Tanner graph for a simple example is shown in Figure 25. This graph corresponds to the following parity check matrix.

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{98}$$

Note that u_1 has neighbors v_1, v_3, v_5, and v_7, which corresponds to the locations of ones in the first row of $\mathbf{H}^{(n'-k)\times n'}$. The relationship holds for the second and third check nodes as well.

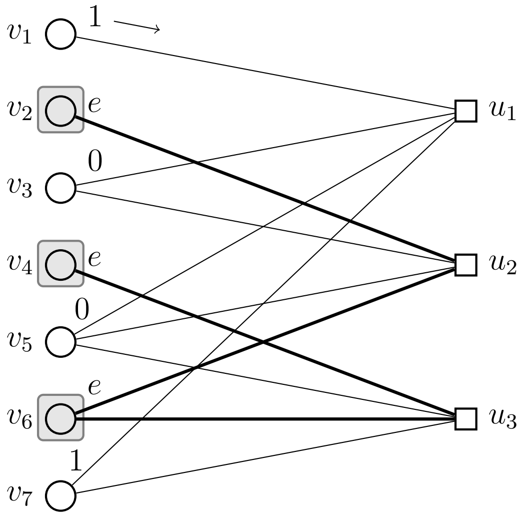

Figure 25: Tanner graph for MP decoding over the BEC with a highlighted stopping set caused by erasures at variable nodes v_2, v_4, and v_6.

We will briefly mention here the existence of two general types of LDPC codes: regular, and irregular.

Definition 12. A *regular* LDPC code is a code in which the number of ones in each row of **H** is constant and equal for all rows. Similarly, the number of ones in each column is equal for all columns.

Definition 13. An *irregular* LDPC code is any LDPC code that is not regular.

To enhance these definitions, we present the idea of the *degree distribution pair* from the edge perspective, $(\lambda(x), \rho(x))$. Let λ_i equal the fraction of edges in the Tanner graph representation of a code that emanate from a variable node of degree i, where the *degree* of a node is simply the number of edges that connect to it—or are *incident* to it. The maximum degree of all variable nodes in the graph is d_v. Similarly, let ρ_i equal the fraction of edges the graph that emanate from check nodes with degree i, and call the maximum degree of all check nodes d_c. Then we can write down the degree distribution pair from the edge perspective in polynomial form using the equations [12]

$$\lambda(x) = \sum_{i=2}^{d_v} \lambda_i x^{i-1}, \tag{99}$$

and

$$\rho(x) = \sum_{i=2}^{d_c} \rho_i x^{i-1}. \tag{100}$$

One can also write the degree distribution from the node perspective where coefficients in the polynomials simply represent the fraction of nodes of a certain degree. The interested reader is referred to [12] for more information.

Decoding of an LDPC codeword over a binary erasure channel (BEC) can be accomplished using maximum-likelihood (ML) decoding [59], by solving a system of equations. However, the iterative message-passing (MP) decoder is commonly used because of its computational efficiency. We briefly explain both decoders, after fully defining the BEC.

5.3.2 Maximum-Likelihood Decoding over the Binary Erasure Channel

Let us consider an LDPC codeword $\mathbf{c} = (c_1, c_2, \ldots, c_{n'}) \in \mathcal{C}$ transmitted over a binary erasure channel (BEC) and let $\mathbf{y} = (y_1, y_2, \ldots, y_{n'})$ denote the received codeword. Then $c_i \in \{0, 1\}$ and $y_i \in \{0, 1, e\}$ where e signifies an erased bit. Figure 26 gives the graphical representation of the BEC model, where bits are erased at random with probability δ_b, or passed error free with probability $(1 - \delta_b)$.

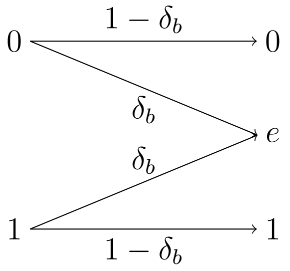

Figure 26: Binary erasure channel model.

Define $\mathcal{K} = \{i : y_i \neq e\}$ and $\bar{\mathcal{K}} = \{i : y_i = e\}$. Furthermore, $\mathbf{H}_{\mathcal{K}}$ and $\mathbf{H}_{\bar{\mathcal{K}}}$ can be understood to be matrices formed by the columns of $\mathbf{H}^{(n'-k)\times n'}$ indexed by $\mathcal{K}$ and $\bar{\mathcal{K}}$, respectively. Similarly, $\mathbf{c}_{\mathcal{K}}$ and $\mathbf{c}_{\bar{\mathcal{K}}}$ are vectors composed of only the bits indexed by the respective sets $\mathcal{K}$ and $\bar{\mathcal{K}}$.

Clearly,

$$\mathbf{H}\mathbf{c}^T = \mathbf{H}_{\mathcal{K}}\mathbf{c}_{\mathcal{K}}^T + \mathbf{H}_{\bar{\mathcal{K}}}\mathbf{c}_{\bar{\mathcal{K}}}^T = 0, \tag{101}$$

where $\mathbf{c}_{\mathcal{K}} = \mathbf{y}_{\mathcal{K}}$, and thus,

$$\mathbf{H}_{\mathcal{K}}\mathbf{c}_{\mathcal{K}}^T = \mathbf{z}^T \tag{102}$$

is known. The ML decoder must then solve for the channel-erased bits $\mathbf{c}_{\bar{\mathcal{K}}}$ using the system of equations given by

$$\mathbf{H}_{\bar{\mathcal{K}}}\mathbf{c}_{\bar{\mathcal{K}}}^T = \mathbf{z}^T. \tag{103}$$

This system has a unique solution when the erased bits are such that the columns of $\mathbf{H}_{\bar{\mathcal{K}}}$ are linearly independent [60]. We can obtain a bound from this statement that we will use to analyze security in the worst-case.

Proposition 1. *For a linear code C with blocklength n' and dimension k, the ML decoder over the BEC cannot have a unique solution if the number of erasures exceeds $(n'-k)$, that is if $|\bar{\mathcal{K}}| > n'-k$.*

Proof. The rank of $\mathbf{H}_{\bar{\mathcal{K}}}$ equals the number of linearly independent rows or columns of the matrix ([61], pg. 244), and cannot exceed the height of $\mathbf{H}_{\bar{\mathcal{K}}}$, which is $n'-k$. □

In fact, when the number of erasures exceeds $(n'-k)$, the system in (103) will be such that the degrees of freedom in the ML decoder

$$D_{ML} \geq |\bar{\mathcal{K}}| - (n'-k), \tag{104}$$

where we achieve equality if the rank of $\mathbf{H}_{\bar{\mathcal{K}}}$ equals $(n'-k)$ [59]. In any case,

$$D_{ML} = \max\{|\bar{\mathcal{K}}| - \mathrm{rank}(\mathbf{H}_{\bar{\mathcal{K}}}), 0\}. \tag{105}$$

This definition clearly satisfies the notion of degrees of freedom from Definition 11 for this decoder. Thus we see, that the effectiveness of the decoder is strictly bounded by the redundancy of the code. While faster methods have been discovered for solving a linear system of equations, the straightforward decoder is known to have complexity $((1-R)\beta+\gamma\delta)\delta^2(n')^3$, where R is the rate of the code, β and γ are constants that are also a function of the elimination algorithm chosen to solve the system of equations, δ is the erasure probability in the channel, and n' is the blocklength of the code [59].

5.3.3 Message-Passing Decoding

The MP decoder is an iterative decoder that can be understood easily using the Tanner graph representation of $\mathcal{C}$. The decoding process passes *messages* between U and V along the edges of $G_{\mathcal{C}}$. One version of the decoder is given as Algorithm 1 (adapted from [60]). The number of degrees of freedom in the MP decoder D_{MP} is the cardinality of the smallest set of bit values that must be supplied to decode all remaining bits. If the decoder succeeds, then $D_{MP} = 0$. Clearly, this maintains the definition of degrees of freedom given in Definition 11 when restricted to this decoder, because any bit combination of these D_{MP} values decodes to a valid codeword, and each is equally likely without further information. A bound on the correction capabilities of the MP decoder is given by the following proposition.

Proposition 2. *The MP decoder over the BEC can correct no more than $(n'-k)$ erasures.*

Proof. In Algorithm 1, each check node can correct at most one variable node, and $|U| = n' - k$. □

The MP decoder is suboptimal compared with the ML decoder, although the MP decoder has linear complexity in the blocklength [12]. A more detailed comparison of the two decoders is offered in [62].

Algorithm 1 MP Decoder over the BEC [60].

```
1: **Initialize:** For $y_i \neq e$, set $v_i = y_i$ and $A = \{v_i : y_i \neq e\}$.
2: **if** ($A = \emptyset$ and no check node has degree one) **then**
3:     Output the (possibly partial) codeword and stop.
4: **else**
5:     Delete all $v_i \in A$ and their adjacent edges.
6: **end if**
7: For each $v_j$ connected to a degree one check node $u_i$, set $v_j = \sum_{k \in \mathcal{N}_{i,j}} v_k$, and
   assign $v_j$ to $A$. Jump to 2.
```

5.3.4 Stopping Sets

To make D as large as possible for our system when an eavesdropper uses an MP decoder, we would like to design the encoder block from Figure 24 so that every bit erased by the channel adds a degree of freedom to the decoder. Stopping sets provide a means of accomplishing this task.

Definition 14 (Di, et. al. [63])**.** A *stopping set* is a set $S \subseteq V$ such that all check nodes in $N(S)$ are connected to S by at least two edges, where $N(S)$ signifies the *neighborhood* of S and is defined as the set of all adjacent nodes to any member of S in $G_{\mathcal{C}}$.

The empty set, by definition, is a stopping set, as is any union of stopping sets. Thus, any set of variable nodes has a unique maximal stopping set in it.[1] See Figure

[1] We will sometimes ignore the empty set as a stopping set and say that a set A *contains no stopping sets*, meaning that the maximal stopping set in A is $\emptyset$.

25 for a simple example; clearly the erasures cannot be resolved using Algorithm 1. This gives way to the following lemma.

Lemma 4 (Di et. al. [63], Lemma 1.1)**.** *Let $G_{\mathcal{C}}$ be the Tanner graph defined by the parity check matrix $\mathbf{H}$ of a binary linear block code $\mathcal{C}$, and assume that $\mathcal{C}$ is used to transmit over the BEC. Let A be the set of erased bits in the received codeword. Then, using Algorithm 1 on $G_{\mathcal{C}}$, the set of erasures that remain after decoding comprise the unique maximal stopping set in A.*

Since stopping sets cause the MP decoder to fail, puncturing in the encoder is done with the intent to inflict Eve with stopping sets. However, the ML decoder will still succeed, even in the presence of stopping sets, as long as the erased bits have linearly independent columns in $\mathbf{H}$. We account for both decoders in our design by using a particular ensemble of LDPC codes where D_{MP} can be made equal to D_{ML}, thus ensuring secrecy even if Eve chooses to use an ML decoder. This issue is addressed in Sections 5.4.3 and 5.4.4. The simplicity of MP decoding is also preserved for all legitimate receivers.[2]

5.4 Encoder

The encoder design is based on the fact that $\mathbb{I}(\mathbf{E};\mathbf{Z}) \leq \mathbb{I}(\mathbf{E};\mathbf{X})$ because processing cannot increase information, and $\mathbf{E} \rightarrow \mathbf{X} \rightarrow \mathbf{Z}$ is a Markov process [13]. The key idea in the decoder is to reduce the bits in $\mathbf{X}$ to the decoding threshold. In other words, $\mathbf{X}$ can be used to recover $\mathbf{E}$ by design, but any erased bit over Q_w makes unique decodability impossible. Proper design maximizes D for Eve. The stages of encoding are portrayed in Figure 27, where each stage fulfills a specific purpose within the overall goals of obtaining secrecy and reliability. The following principles are addressed in the design of this encoder.

[2] For further information on stopping sets as they relate to LDPC code ensembles, see [64] and [65].

- Bits of $\mathbf{E}$ are hidden from immediate access in the decoded codewords using nonsystematic LDPC codes.

- Scrambling prior to coding magnifies errors in the decoder.

- The error-correction capabilities of the LDPC code are restricted by intentional puncturing of encoded bits. (Bob obtains reliability through ARQ, rather than error correction.)

- Bits from encoded blocks are interleaved throughout all packets so that an erased packet results in stopping sets in all codewords.

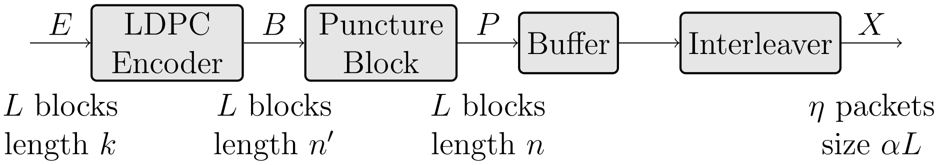

Figure 27: Detailed block diagram of the encoder. Number and size of blocks or packets are indicated at each step.

5.4.1 Nonsystematic LDPC Codes

Recall from Section 5.2 that

$$\mathbf{E} = \mathbf{E}^{kL} = (\mathbf{E}_1^k, \mathbf{E}_2^k, \ldots, \mathbf{E}_L^k), \tag{106}$$

where

$$\mathbf{E}_i^k = (E_{(i-1)k+1}, E_{(i-1)k+2}, \ldots, E_{ik}) \tag{107}$$

for $i = 1, 2, \ldots, L$. These L blocks of encrypted message form the input to the nonsystematic LDPC encoder with blocklength n' and dimension k. The output of the LDPC encoder $\mathbf{B}$ is given as L codewords of length n', denoted as

$$\mathbf{B} = \mathbf{B}^{n'L} = (\mathbf{B}_1^{n'}, \mathbf{B}_2^{n'}, \ldots, \mathbf{B}_L^{n'}), \tag{108}$$

where each vector

$$\mathbf{B}_i^{n'} = (B_{(i-1)n'+1}, B_{(i-1)n'+2}, \ldots, B_{in'}). \tag{109}$$

Certainly, if the code $\mathcal{C}$ is *systematic*, then the bits of $\mathbf{E}_i^k$ appear explicitly in the codeword $\mathbf{B}_i^{n'}$. Therefore, we choose $\mathcal{C}$ to be *nonsystematic*, so the input bits do not appear explicitly in the output bits.

Nonsystematic LDPC coding is typically implemented as a two stage process to improve encoder complexity [66, 67, 31]. Let $\mathbf{S}^{k\times k}$ be an invertible *scrambling* matrix over $GF(2)$, and let $\mathbf{G}$ be a $k \times n'$ systematic generator matrix. The ith block of $\mathbf{E}$, $\mathbf{E}_i^k$, is a length-k encrypted message. Then our LDPC encoder scrambles $\mathbf{E}_i^k$ by the operation

$$\mathbf{A}_i^k = \mathbf{E}_i^k \mathbf{S}^{k\times k}, \tag{110}$$

where $\mathbf{A} = (\mathbf{A}_1^k, \mathbf{A}_2^k, \ldots, \mathbf{A}_L^k)$ is the collection of scrambled data blocks. The data are then encoded using $\mathbf{G}^{k\times n'}$ by the operation

$$\mathbf{B}_i^k = \mathbf{A}_i^k \mathbf{G}^{k\times n'} \tag{111}$$

for $i = 1, 2, \ldots, L$ to obtain $\mathbf{B}$.

Clearly, the inverse operation at the decoder first requires the bits of $\mathbf{B}$ to be obtained through either MP or ML decoding. Since $\mathbf{G}$ is systematic, the bits of $\mathbf{A}$ are explicit in $\mathbf{B}$. The bits of $\mathbf{E}$ can then be found by applying the inverse of $\mathbf{S}^{k\times k}$ a block at a time in the descrambling operation

$$\mathbf{E}_i^k = \mathbf{A}_i^k \mathbf{S}^{-1}. \tag{112}$$

This process amplifies errors in the decoding process as a function of the sparsity of $\mathbf{S}^{-1}$. We find $\mathbf{S}^{-1}$ using the LU decomposition [61] over $GF(2)$. By simulation results, we know that nonsingular randomly generated scrambling matrices have inverses with just less than 50% of the entries equal to one on average. Thus, the resulting descrambling operation in (112) is enough to cause even a single error in $\mathbf{A}_i^k$

to propagate into a bit-error rate (BER) of 0.5 in $\mathbf{E}_i^k$ (see Section 5.8). This result is intuitive because a bit in $\mathbf{E}_i^k$ is a linear combination of bits in $\mathbf{A}_i^k$. Say that a single bit E_j is the $GF(2)$ sum of several bits in $\mathbf{A}$, due to the descrambling operation, so that

$$E_j = A_{l_1} \oplus A_{l_2} \oplus \cdots \oplus A_{l_\xi}. \tag{113}$$

Then, E_j is in error if an odd number of bits in the set $\{A_{l_1}, A_{l_2}, \ldots, A_{l_\xi}\}$ are in error. On average, the row weights in $\mathbf{S}^{-1}$ are approximately $k/2$, and the expectation of $k/2$ bits in error holds for any number of errors in $\mathbf{A}_i^k$.

Since only one $(\mathbf{S}, \mathbf{S}^{-1})$ pair is needed, the matrices can be generated offline. The complexity of both the encoder and decoder is increased by the matrix multiplications in (110) and (112), respectively. Both of these operations are $\mathcal{O}(k^3)$. General systematic encoder complexity is $\mathcal{O}((n')^2)$ because G is not sparse by design [12], although improvements can be made using appropriate preprocessing as outlined in [60]. The encoding technique specified in [60] gives encoder complexity of $\mathcal{O}(n' + g^2)$ where g is the *gap* in an approximate lower triangular form of the parity check matrix and is less than $n' - k$. The complexities for the ML and MP decoders are given in Sections 5.3.2 and 5.3.3 as $\mathcal{O}((n')^3)$ and $\mathcal{O}(n')$, respectively.

5.4.2 Puncturing

The next step is to puncture bits from each codeword in $\mathbf{B}$. Let the puncturing pattern $R \in V$ be the set of bits to be punctured in each $\mathbf{B}_i^{n'}$. Recall that V is the set of variable nodes in the Tanner graph $G_{\mathcal{C}}$. The punctured blocks

$$\mathbf{P} = \mathbf{P}^{nL} = (\mathbf{P}_1^n, \mathbf{P}_2^n, \ldots, \mathbf{P}_L^n), \tag{114}$$

where each

$$\mathbf{P}_i^n = (P_{(i-1)n+1}, P_{(i-1)n+2}, \ldots, P_{in}), \tag{115}$$

are shown in Figure 27 to have length n, which was defined in Section 5.2 to be the effective blocklength of the entire encoder. All unpunctured bits belong to the set Q

so that $V = R + Q$; therefore, the length of each block in $\mathbf{P}$ is equal to $|Q| = n$. The puncturing pattern R is chosen to induce stopping sets in an eavesdropper's received data.

Definition 15. A puncturing pattern R is *acceptable* if and only if there are no stopping sets in R, and $R + v$ contains some nonempty stopping set S_v for every $v \in Q$.

Such a set R can be constructed using the random technique outlined in Algorithm 2, which also calls Algorithm 3 to check for stopping sets in a computationally tractable manner [9]. For Algorithm 3 to make sense, we must define the notion of an induced subgraph.

Definition 16. Let G be a graph with a set of vertices $V = \{v_1, v_2, \ldots, v_{n'}\}$ and a set of edges E. Denote the edge between vertices v_i and v_j as $v_i v_j$. Furthermore, consider a subset of vertices $V' \subseteq V$. Then the graph that has V' as its vertex set, and exactly those edges $xy \in E$ such that $x, y \in V'$ as its edge set, is an *induced subgraph* of G. We say that V' induces such a graph on G and write

$$G' =: G[V'] \tag{116}$$

to indicate the induced subgraph G' [68].

Algorithm 2 Finds an acceptable puncturing pattern $R \subseteq V$.

```
Initialize: $R = v$, for some $v \in V$, $Q = \emptyset$.
while ($V \backslash (R \cup Q) \neq \emptyset$) do
    Check for stopping sets in $R + v$ for another randomly chosen $v \in V \backslash (R \cup Q)$.
    if ($R + v$ has a stopping set) then
        $Q = Q + v$.
    else
        $R = R + v$.
    end if
end while
```

Algorithm 3 Checks for nonempty stopping sets in $A \subseteq V$ [9].

```
Initialize: $S = A$
while ($S \neq \emptyset$) do
    Induce the subgraph $G' = G[S \cup N(S))]$.
    if ($\exists$ a check node in $G'$ with degree 1) then
        Delete variable nodes in $S$ that are adjacent to check nodes of degree 1
        in $G'$.
    else
        Return true. $S$ is the maximal nonempty stopping set in $A$.
    end if
end while
Return false. There is no nonempty stopping set in $A$.
```

Lemma 5. *The output of Algorithm 2 is always an* acceptable *puncturing pattern R as defined in Definition 15.*

Proof. We must first show that upon completion of Algorithm 2, there are no stopping sets in R. Assume for a contradiction that R has a stopping set. Then there is a bit $v \in R$ that when added to R during the construction process, caused a stopping set to first appear. Then by Algorithm 2, $v \notin R$. This provides the contradiction. It remains to be proved that Algorithm 3 operates as expected.

Proposition 3. *Algorithm 3 always returns true when A has a nonempty stopping set, and always returns false otherwise.*

Proof of Proposition. Suppose that the bits in A are erasures over the BEC, and Algorithm 1 is used to decode. Realize that erasures recovered in the ith iteration of Algorithm 1 correspond exactly to the nodes deleted in the ith iteration of Algorithm 3. If all bits can be resolved using MP decoding then all nodes will be deleted in Algorithm 3, and false is returned. If, however, MP decoding returns a partial codeword, then Algorithm 3 will return true because all remaining bits have degree greater than one in the induced subgraph G'. By Lemma 4, the remaining nodes comprise the maximal stopping set of A. □

To complete the proof of Lemma 5, we must also show that for any $v \in Q$, $R + v$

has a nonempty stopping set. Since in Algorithm 2 every $v \in Q$ is such that for some subset $R' \subseteq R$, $R' + v$ has a stopping set, therefore $R + v$ has a stopping set for any $v \in Q$. $\square$

Thus, puncturing according to R in each $\mathbf{B}_i^{n'}$ for $i = 1, 2, \ldots, L$, guarantees that every bit in each $\mathbf{P}_i^n$ is crucial for successful MP decoding.

Complexity of Algorithm 2 is linear in the blocklength n', because it chooses $(n' - 1)$ bits in a random order, and calls Algorithm 3 after each choice. The complexity of Algorithm 3 in the worst case, is quadratic in $|U| = (n' - k)$, the number of check nodes in $G_{\mathcal{C}}$. Line 5 of the algorithm will be repeated a maximum of $\sum_{i=1}^{|U|} i = \frac{|U|^2 + |U|}{2}$ times if a single node is deleted each time the line is executed. Therefore, the complexity of finding R is at most quadratic in $|U|$, and linear in n', i.e. has complexity $\mathcal{O}(n'|U|^2)$. As a result, the algorithm can be used in practical system design to compute R offline.

5.4.3 Regular versus Irregular Codes

The overall rate k/n of the nonsystematic and punctured code is a function of the rate of the systematic LDPC code, and $|R|$. Simulations have shown that the size of R is a function of the degree distribution on C, although the exact relationship is still unknown.

Example 4. Let C be a regular rate-1/2 code with $n' = 1000$, $w_c = 4$, and $w_r = 8$, where w_c and w_r are the fixed column and row weights of the parity check matrix, respectively. The size of $|R|$ appears to be Gaussian-distributed for this family of codes with a mean size of approximately 436, with variance roughly equal to 15. Let us examine, however, an irregular ensemble with the same rate and blocklength, but having the following degree distribution pair from the edge perspective: $\lambda(x) = 0.32660x + 0.11960x^2 + 0.18393x^3 + 0.36988x^4$ on variable node weights, and $\rho(x) = 0.78555x^5 + 0.21445x^6$ on check node weights (see (99) and (100), as well as [12] pg. 664), where $\mathbf{H}$ is formed using the socket approach given in [59]. Here the distribution on $|R|$ is much tighter, ranging from 496 to 500. The cardinality of R is equal to 500 with probability approximately 0.1, 499 with probability roughly 0.56, and 498 with probability near 0.26. Thus, codes exist for which Algorithm 2 returns R such that $|R| = n' - k$ with reasonable probability.

As a direct result, a puncturing pattern generated for the irregular code in Example 4 has a unique property. That is, for some patterns, $D_{MP} = D_{ML}$.

Lemma 6. *Let R_c denote the indices of the channel-erased bits of $\mathbf{P}_i^n$, and D_{MP} and D_{ML} denote the degrees of freedom using MP decoding and ML decoding, respectively. If an irregular LDPC code is employed over the BEC with intentional puncturing such that $|R| = n' - k$, then $D_{ML} = D_{MP} = |R_c|$.*

Proof. By Propositions 1 and 2, the respective ML and MP decoders can correct a maximum of $n' - k$ erasures. Since $|R| = n' - k$, any erasure by the channel is guaranteed to give a degree of freedom in either decoder. □

It should be noted that if the sum of systematic bits in $R + R_c$ is less than D, a brute-force attack on these bits might be more appealing to an attacker than decoding the entire codeword. To cover this possibility, D should be defined as the minimum between the number of systematic bits missing to the eavesdropper, and the degrees of freedom in the decoder. Although, in practice the number of systematic bits removed through puncturing or erased by the channel exceeds the degrees of freedom in the decoder with high probability.

5.4.4 Interleaving

The role of the interleaver is to ensure that all packets must be obtained error free for successful decoding in any and all encoded blocks. To do this, we construct a collection of η packets to be transmitted $\mathbf{X} = (\mathbf{X}_1^{\alpha L}, \mathbf{X}_2^{\alpha L}, \dots, \mathbf{X}_\eta^{\alpha L})$ in the following manner. Alice picks α to be a small positive integer that is assumed to divide n for ease in notation. Then, $\eta = \frac{n}{\alpha}$ and the ith packet is formed as

$$\begin{aligned}\mathbf{X}_i^{\alpha L} =& (P_{(i-1)\alpha+1}, \dots, P_{i\alpha}, P_{(i-1)\alpha+n+1}, \dots, P_{i\alpha+n}, \dots, \\ & P_{(i-1)\alpha+(n-1)n+1}, \dots, P_{i\alpha+(n-1)n})\end{aligned} \tag{117}$$

for $i = 1, 2, \dots, \eta$. Although this expression is accurate, it is admittedly cumbersome. In words, we form the packet $\mathbf{X}_i$ by concatenating α bits from each encoded and punctured block P_j for $j = 1, 2, \dots, L$. Therefore, a single erased packet causes α channel erasures in each punctured block at the decoder. Since we have designed R so that any erasure results in a stopping set, we can be assured that any erased packet will cause all L blocks to fail in the MP decoder. If $|R| = n' - k$, then the same result holds for ML decoding by Lemma 6.

Corollary 1. *If $|R| = n' - k$ and packets are formed according to (117), then the number of degrees of freedom in the ith codeword is $D_{ML}^i = D_{MP}^i = |R_c^i| = \alpha|R_p|$ for $i = 1, 2, \dots, L$, where R_p is a list of all erased packets and R_c^i are the channel-erased bits in the ith codeword. Furthermore, $D_{ML}^i = D_{MP}^j \forall i, j$.*

Proof. The first part is trivial and follows directly from Lemma 6 and (117). We see that $D_{ML}^i = D_{MP}^j$ because a missing packet means exactly α degrees of freedom in each block, irrespective of decoder choice. □

5.5 Decoder for Legitimate Users

The decoder for legitimate users is simply the inverse of all encoder operations. A user can decode all data as long as every packet is received error free. Legitimate users make use of the authenticated feedback channel to request retransmission of packets erased in the main channel during transmission.[3] The decoding process is depicted in Figure 28. Once all packets are obtained in $\tilde{\mathbf{Y}}$, the bits are deinterleaved

[3] Time delay and queueing aspects of ARQ protocols are well-addressed in the literature, e.g. [69] and its references.

back into their intentionally punctured codewords $\tilde{\mathbf{P}}$. The MP decoder is guaranteed to decode in linear time with the blocklength to obtain $\tilde{\mathbf{B}}$ [12], and the inverse of the scrambling matrix is applied to the systematic decoded bits using (112) to obtain $\tilde{\mathbf{E}}$. Once all packets are known, this decoder guarantees that $\tilde{\mathbf{E}} = \mathbf{E}$.

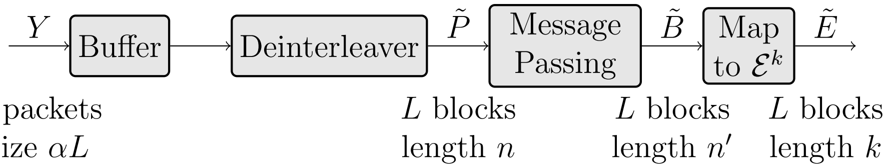

Figure 28: Detailed block diagram of Bob's decoder. Number and size of blocks or packets are indicated at each step.

5.6 Security against Wiretappers

An eavesdropper can decode the data using Bob's decoder in Figure 28 if all packets are obtained error free. The independence of Q_m and Q_w, however, prevents Eve from receiving packets as a function of δ and ϵ, the respective probabilities of erasures in Q_m and Q_w. Let R_{ef} be the event that a single packet is *received error free* by at least one eavesdropper after all retransmissions of the packet requested by any legitimate receiver have been filled. This section shows the blanket security effect of our encoder over nearly the entire region of possible (δ, ϵ) pairs by characterizing the distribution on D. General security results are shown as a function of $\Pr(R_{ef})$, followed by expressions for $\Pr(R_{ef})$ for the wiretap channel case, the broadcast scenario with m intended receivers, the case with l collaborating eavesdroppers, and the most general case with both m legitimate receivers and l collaborating eavesdroppers. Legitimate receivers are always given access to the feedback channel, and eavesdroppers are always restricted to passive status, using authentication. Retransmissions in the ARQ protocol are executed only after requests are received from all legitimate parties. The results assume an encoder that satisfies Corollary 1; therefore, D represents the degrees of freedom in every codeword using either the ML or MP decoder.

5.6.1 General Security Theorems

Lemma 7. *The random variable D that governs the number of degrees of freedom in a received codeword is a scaled binomial random variable. Thus, for $1 \leq \beta \leq \alpha\eta$,*

$$\Pr(D \geq \beta) = 1 - \sum_{i=0}^{\lceil \beta/\alpha \rceil - 1} \binom{\eta}{i} (1 - \Pr(R_{ef}))^i \Pr(R_{ef})^{\eta - i}. \tag{118}$$

Proof. By definition of R_{ef}, packets are erased for eavesdroppers with probability $(1 - \Pr(R_{ef}))$. Since there are η independent Bernoulli trials, each identically distributed, the sum of erased packets $|R_p|$ is a binomial random variable with parameters η and $(1-\Pr(R_{ef}))$ [70]. Then, by Corollary 1, $D = \alpha|R_p|$ where α bits from every codeword are sorted into each packet. Thus, D is a scaled binomial random variable; specifically $D \sim \text{Bin}(\eta, 1 - \Pr(R_{ef}))\alpha$. Since $D = \alpha|R_p|$, then $D \geq \beta$ implies that $\alpha|R_p| \geq \beta$. Clearly, this requires that $|R_p| \geq \lceil \beta/\alpha \rceil$. The result in (118) follows directly. □

The expected value is, therefore, known because of the binomial structure of D. We also prove an important property in regards to $\mathbb{E}[D]$ in the following theorem.

Theorem 5. *If $|R| = n' - k$ in the encoder, then $\frac{k}{n} = 1$, and*

$$\mathbb{E}[D] = \frac{1}{L}\mathbb{H}(\mathbf{X}|\mathbf{Z}) = (1 - \Pr(R_{ef}))n = (1 - \Pr(R_{ef}))k. \tag{119}$$

Proof. Since $|R| = n' - k$, then $n = |Q| = n' - |R| = k$. Therefore, $\frac{k}{n} = 1$. Furthermore, we can assume η independent uses of a PEC with packets of length αL. Let $\mathbf{X} = (\mathbf{X}_1^{\alpha L}, \mathbf{X}_2^{\alpha L}, \ldots, \mathbf{X}_\eta^{\alpha L})$ be the input to the channel, and $\mathbf{Z} = (\mathbf{Z}_1^{\alpha L}, \mathbf{Z}_2^{\alpha L}, \ldots, \mathbf{Z}_\eta^{\alpha L})$ be the output, where αL bits are erased with probability $(1 - \Pr(R_{ef}))$ or received error free with probability $\Pr(R_{ef})$ with each channel use. The input distribution on αL bits in each packet is uniform because the input distribution on E is uniform, and the encoding function with rate $\frac{k}{n} = 1$ forms a bijection on k bits. Thus, $\mathbb{H}(\mathbf{X}_i) = \alpha L$ for $i = 1, 2, \ldots, \eta$. It can be shown that $\mathbb{H}(\mathbf{Z}_i|\mathbf{X}_i) = \mathbb{H}(1 - \Pr(R_{ef}))\alpha L$, and $\mathbb{H}(\mathbf{Z}_i) = \mathbb{H}(1 - \Pr(R_{ef}))\alpha L + \Pr(R_{ef})\alpha L$ (see [13], pg. 188). Then,

$$\begin{aligned} \frac{1}{L}\mathbb{H}(\mathbf{X}_i|\mathbf{Z}_i) &= \frac{1}{L}\left(\mathbb{H}(\mathbf{Z}_i|\mathbf{X}_i) - \mathbb{H}(\mathbf{Z}_i) + \mathbb{H}(\mathbf{X}_i)\right) \\ &= \alpha(1 - \Pr(R_{ef})). \end{aligned} \tag{120}$$

Therefore, with η independent uses of the channel (one for each packet),

$$\begin{aligned} \frac{1}{L}\mathbb{H}(X|Z) &= (1 - \Pr(R_{ef}))\eta\alpha \\ &= (1 - \Pr(R_{ef}))n. \end{aligned} \tag{121}$$

Since the mean of a binomial random variable is the product of its two parameters, then

$$\mathbb{E}\left[\frac{D}{\alpha}\right] = (1 - \Pr(R_{ef}))\eta, \tag{122}$$

which implies the equivalence stated in the theorem. □

Thus we see that the expected number of degrees of freedom in each codeword, $\mathbb{E}[D]$, is equal to the per-codeword equivocation when the puncturing is accomplished so that $|R| = n' - k$. Perfect secrecy is then obtained when $\mathbb{E}[D] = k$, because $\frac{1}{L}\mathbb{H}(\mathbf{X}) = k$. Of course, this occurs when $\Pr(R_{ef}) = 0$, which implies that the eavesdropper obtains zero packets. Thus, this scheme cannot achieve perfect secrecy in practice. However, it can be shown using the achievable rates in [21] that $\mathbb{E}[D]$ approaches the maximum achievable equivocation when $\frac{k}{n} = 1$.

The encoder design is such that an eavesdropper will likely need to resolve the degrees of freedom through trial and error. The number of expected trials to guess D degrees of freedom in a single codeword was given in (92) as

$$\mathbb{E}[\frac{1}{2}(2^D + 1)] = \frac{1}{2}(\mathbb{E}[2^D] + 1).$$

Now that the distribution on D is known, $\mathbb{E}[2^D]$ can be calculated. Let us assume that $\alpha = 1$, and therefore, $\eta = n$ for simplicity of notation. With $\alpha = 1$, the pmf of

D is

$$\Pr(D = d) = \binom{n}{d} (1 - \Pr(R_{ef}))^d \Pr(R_{ef})^{n-d}. \tag{123}$$

Therefore, the expection of 2^D is given as

$$\begin{aligned}\mathbb{E}[2^D] &= \sum_{d=0}^{n} 2^d \binom{n}{d} (1 - \Pr(R_{ef}))^d \Pr(R_{ef})^{n-d} \\ &= \sum_{d=0}^{n} \binom{n}{d} [2(1 - \Pr(R_{ef}))]^d \Pr(R_{ef})^{n-d} \\ &= (2 - \Pr(R_{ef}))^n. \end{aligned} \tag{124}$$

The last step is calculated using the well-known binomial theorem [70]

$$\sum_{i=0}^{n} \binom{n}{i} a^{n-i} b^i = (a + b)^n. \tag{125}$$

Therefore, in a general sense, by applying (92) we can anticipate that an attacker must make an average of

$$\frac{1}{2}[(2 - \Pr(R_{ef}))^n + 1] \tag{126}$$

guesses to resolve D degrees of freedom in a single codeword ($L = 1$).

5.6.2 Bounds on Security Enhancement for General Ciphers

We now generate bounds on attack complexity against general ciphers when a transmission includes multiple codewords ($L \geq 1$), and $D = d$ degrees of freedom are present in an attacker's received codewords. Since an attack against the underlying cryptography could feasibly be staged using a single block of $\hat{\mathbf{E}}$, especially with large blocklengths, we only guarantee failure of the attack if *every* block in $\hat{\mathbf{E}}$ has an expected error rate of 0.5. Using similar logic, it can be said that if an attack would succeed using the error-free ciphertext $\mathbf{E}$, then it *may* fail even if a single block in $\hat{\mathbf{E}}$ has an error rate of 0.5. Note that D is fixed in this analysis, but an approximation of the security for random D can be obtained by taking the expectation of the bounds.

Before we present the bounds, we must take an aside to calculate the expectation of the minimum of L uniform discrete random variables.

Lemma 8. *Let $U_1, U_2, \ldots, U_L$ be uniform discrete random variables over the alphabet $\mathcal{U} = \{1, 2, \ldots, n_u\}$. Then*

$$\begin{aligned}\mathbb{E}[\min(U_1, U_2, \ldots, U_L)] &= \frac{1}{n_u^L} \sum_{i=1}^{n_u} i^L \\ &= \frac{1}{n_u^L (L+1)} \sum_{i=0}^{L} \binom{L+1}{i} B_i n_u^{L+1-i}, \end{aligned} \tag{127}$$

where B_i is the ith Bernoulli number, and can be calculated recursively with $B_0 = 1$ and

$$B_i = 1 - \sum_{k=0}^{i-1} \binom{i}{k} \frac{B_k}{i-k+1}. \tag{128}$$

Proof. The last equality in (127) follows from Faulhaber's formula [71]

$$\sum_{i=1}^{n_u} i^L = \frac{1}{(L+1)} \sum_{i=0}^{L} \binom{L+1}{i} B_i n_u^{L+1-i}, \tag{129}$$

where $B_1 = +\frac{1}{2}$. For proof of the first equality, let $Y = \min(U_1, U_2, \ldots, U_L)$, and let us note the expression

$$\Pr(Y > y) = \left(\frac{n_u - y}{n_u}\right)^L \tag{130}$$

and use it to calculate the pmf of Y

$$\begin{aligned} p_Y(y) = \Pr(Y = y) &= \Pr(Y > y-1) - \Pr(Y > y) \\ &= \left(\frac{n_u - (y-1)}{n_u}\right)^L - \left(\frac{n_u - y}{n_u}\right)^L. \end{aligned} \tag{131}$$

Now the expectation on Y follows as

$$\begin{aligned} \mathbb{E}[Y] &= \sum_{y=1}^{n_u} y p_Y(y) \\ &= 1 \cdot \left[1 - \left(\frac{n_u - 1}{n_u}\right)^L\right] + 2 \cdot \left[\left(\frac{n_u - 1}{n_u}\right)^L - \left(\frac{n_u - 2}{n_u}\right)^L\right] + \cdots + n_u \cdot \left(\frac{1}{n_u}\right)^L \\ &= \frac{1}{n_u^L} \sum_{i=1}^{n_u} i^L. \end{aligned} \tag{132}$$

This completes the proof. □

The expectation of the maximum of L uniform discrete random variables can be calculated in a similar fashion.

Lemma 9. *Let $U_1, U_2, \ldots, U_L$ be uniform discrete random variables over the alphabet $\mathcal{U} = \{1, 2, \ldots, n_u\}$. Then*

$$\mathbb{E}[\max(U_1, U_2, \ldots, U_L)] = n_u + 1 - \mathbb{E}[\min(U_1, U_2, \ldots, U_L)]. \tag{133}$$

Proof. Let $Z = \max(U_1, U_2, \ldots, U_L)$, and let us write down the cdf of Z

$$\Pr(Z \le z) = \left(\frac{z}{n_u}\right)^L. \tag{134}$$

Now, calculate the pmf of Z as before

$$\begin{aligned} p_Z(z) = \Pr(Z = z) &= \Pr(Z \le z) - \Pr(Z \le z-1) \\ &= \left(\frac{z}{n_u}\right)^L - \left(\frac{z-1}{n_u}\right)^L. \end{aligned} \tag{135}$$

The expectation on Z follows easily as

$$\begin{aligned} \mathbb{E}[Z] &= \sum_{z=1}^{n_u} z p_Z(z) \\ &= 1 \cdot \left(\frac{1}{n_u}\right)^L + 2 \cdot \left[\left(\frac{2}{n_u}\right)^L - \left(\frac{1}{n_u}\right)^L\right] + \cdots + n_u \cdot \left[1 - \left(\frac{n_u - 1}{n_u}\right)^L\right] \\ &= n_u - \frac{1}{n_u^L} \sum_{i=1}^{n_u - 1} i^L \\ &= n_u - \mathbb{E}[\min(U_1, U_2, \ldots, U_L)] + 1. \end{aligned} \tag{136}$$

□

With these pieces in place, we state the following theorem giving bounds of secrecy for our encoder over attacking eavesdroppers for fixed D.

Theorem 6. *Define the complexity of a cryptographic attack to be C_A. Let $D = d$ be the degrees of freedom of each of L blocks in $\hat{\mathbf{B}}$. If an attacker must recover at least* *one block of error-free ciphertext to stage a successful attack against the cryptography, then the expected complexity C_{PL} of a successful attack on the system is bounded as*

$$f_C(2^d, L)C_A \leq C_{PL} \leq \left[2^d + 1 - f_C(2^d, L)\right] C_A, \tag{137}$$

where

$$f_C(a, b) = \frac{1}{a^b(b+1)} \sum_{i=0}^{b} \binom{b+1}{i} B_i a^{b+1-i} \tag{138}$$

and B_i is the ith Bernoulli number as in Lemma 8.

Proof. By Corollary 1 each codeword in $\hat{\mathbf{B}}$ has the same number of degrees of freedom. Thus, d bits must be guessed in each of L punctured codewords in $\hat{\mathbf{P}}$. Furthermore, the correctness of a guess must be verified by executing an attack on the underlying cryptography. In this analysis we assume the attack succeeds if the ciphertext is error free, and fails if the ciphertext has BER 0.5. Assume that an attacker can guess bit patterns on all codewords in $\hat{\mathbf{P}}$ simultaneously. The correct bit patterns of the channel-erased bits in the L codewords $\hat{\mathbf{P}}$ are uniformly distributed over 2^d possibilities in each block. The lower bound is formulated by multiplying the attack complexity by the expected number of guesses until at least one of L codewords is found. Thus the lower bound is the product of C_A and the expectation on the minimum of L uniform discrete random variables that are drawn from the alphabet $\mathcal{U} = \{1, 2, \ldots, 2^d\}$. Application of the result from Lemma 8 given in (127) completes the proof of the lower bound.

The upper bound is calculated similarly, but we assume that *all* patterns must be guessed to guarantee success; therefore, the bound is given by the product of C_A and the expectation of the maximum of L uniform discrete random variables from Lemma 9 given in (133). □

More than likely, a successful attack against the cryptography will require at least a certain number of consecutive blocks of error-free ciphertext to execute successfully

[48]. Clearly a 0.5 BER in any block would destroy an attack with these requirements. Therefore, the upper bound in (137) serves as a good approximation to the expected amount of work necessary to complete the attack, with L being set by the amount of ciphertext required for the attack. Thus we see, that our system appends a multiplier that is exponential in d to the complexity of a cryptographic attack by exploiting channel characteristics at the physical-layer.

5.6.3 One Receiver and One Wiretapper

Although the general security solutions are proved in Sections 5.6.1 and 5.6.2, we still require expressions for $\Pr(R_{ef})$ under a variety of scenarios to complete the security characterization in D. The simplest case matches the setup given in Figure 24, and was originally proved in [9].

Lemma 10. *For the packet erasure wiretap channel with feedback for authenticated ARQ, the probability that Eve obtains a single transmitted packet is given as*

$$\Pr(R_{ef}) = \frac{1-\epsilon}{1-\epsilon\delta}. \tag{139}$$

Proof. Let W be the total number of times that Bob requests a single packet over Q_m before he obtains it error free. Recall that it is erased each time independently with probability δ. Therefore, W is a random variable that takes on the number of total transmissions up to and including the first successful reception of the packet, and is thus geometrically distributed with success parameter $(1-\delta)$ [70]. Then,

$$\Pr\left(W = w\right) = (1-\delta)\delta^{w-1}. \tag{140}$$

Recall that the probability of an erased packet in Q_w is ϵ. Therefore,

$$\begin{aligned}
\Pr\left(R_{ef}\right) &= \sum_{w=1}^{\infty} \Pr(R_{ef}|W=w)\Pr(W=w) \\
&= \sum_{w=1}^{\infty}(1-\epsilon^w)(1-\delta)\delta^{w-1} \\
&= (1-\delta)\sum_{w=1}^{\infty}\left(\delta^{w-1}-\epsilon^w\delta^{w-1}\right) \\
&= \frac{1-\epsilon}{1-\epsilon\delta}.
\end{aligned}$$

□

Intuition of security for the wiretap channel in terms of D can be gained by using the expression for $\Pr(R_{ef})$ in (139) to plot $\Pr(D \geq \beta)$ in (118) for different values of β, α, and η. Figure 29 shows $\Pr(D \geq 1)$ for $\eta = 100$. Note that when $\beta = 1$, α is not required to evaluate (118). This case is provided to emphasize the plateau and falloff regions in the (δ, ϵ) grid for $\Pr(D \geq \beta)$. Throughout the plateau region, stopping sets occur in the MP decoder and the ML decoder has linearly dependent columns in $\mathbf{H}_{\bar{\mathcal{K}}}$ with probability very close to one. The results of Lemmas 7 and 10 give

$$\Pr(D \geq 1) = 1 - \left(\frac{1-\epsilon}{1-\epsilon\delta}\right)^{\eta}. \tag{141}$$

This expression can be examined in the limit as $\eta \to \infty$. It is immediate that except for when $\delta = 1$ or $\epsilon = 0$, $\Pr(D \geq 1)$ goes to one for all (δ, ϵ) pairs as η gets large. From Theorem 5, if $|R| = n' - k$, then $\eta = \frac{n}{\alpha} = \frac{k}{\alpha}$. Clearly η grows with k, n and n'. LDPC codes with blocklength $n' = 10,000$ are deemed practical by today's standards. For $\alpha = 1$ and for a carefully chosen R with size roughly 5000, then $\eta \approx 5000$. This case is shown in Figure 30, where as expected, all nontrivial (δ, ϵ) pairs have $\Pr(D \geq 1) \approx 1$.

But of course, a single degree of freedom is easily guessed in an attack. Let us examine the effects on security with larger β. This perspective is provided in Figure 31, where $\eta = 5000$ and $\beta = 50$ with $\alpha = 1$. As can be seen in the figure, there exists a cutoff region, where (δ, ϵ) pairs within the plateau region experience at least

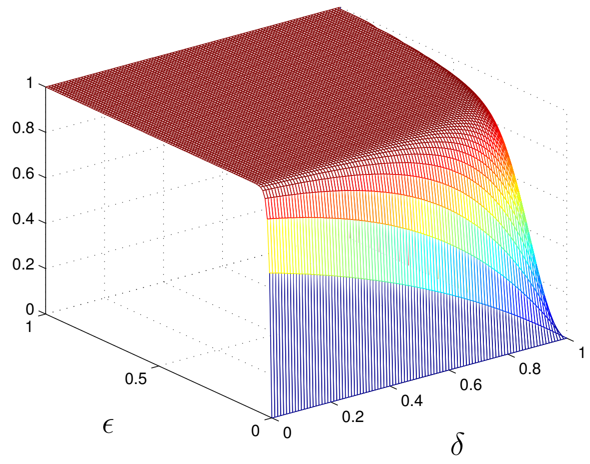

Figure 29: $\Pr(D \geq 1)$ when the number of packets $\eta = 100$, as a function of the respective erasure probabilities in Q_m and Q_w, δ and ϵ.

β degrees of freedom with probability very close to one, while pairs outside the region have $D < \beta$ with probability close to one. Owing to the severity of the cutoff, the threshold can be approximated by setting $\Pr(D \geq \beta) = 0.5$ in (118), and deriving a function of δ and ϵ. This technique provides a unique threshold for each unique triple (β, α, η).

Finally, we calculate $\mathbb{E}[D]$ from Theorem 5 as

$$\mathbb{E}[D] = \frac{\epsilon(1-\delta)}{1-\epsilon\delta}\eta\alpha = \frac{\epsilon(1-\delta)}{1-\epsilon\delta}n. \tag{142}$$

This function grows linearly with n, which is equal to k when $|R| = n' - k$. Thus, to drive D to a large number in practice, we simply increase the dimension of the encoder. Note that in the expectation the choice of α does not affect security; although, $\alpha = 1$ allows η to be as large as possible, thus providing more confidence that single realizations of D are close to $\mathbb{E}[D]$ by the law of large numbers ([70], pg. 193).

5.6.4 Multiple Intended Receivers

In this section, we move past the single user case, and address the more general broadcast channel originally presented in [72]. The model depicts a single eavesdropper with

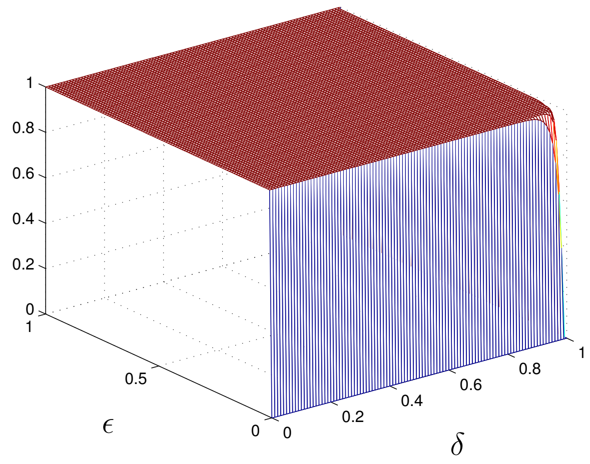

Figure 30: $\Pr(D \geq 1)$ when the number of packets $\eta = 5000$, as a function of the respective erasure probabilities in Q_m and Q_w, δ and ϵ.

probability of an erased packet ϵ, as before. This case allows us to understand the repercussions on security of having more than one user for whom we allow feedback requests. Security is still characterized by Lemma 7 and Theorem 5 in the m user case, but $\Pr(R_{ef})$ must be found. Recall that R_{ef} is the event that Eve receives a single transmitted packet. Let each user have an independent PEC with probability of erasure in the ith user's channel as δ_i for $i = 1, 2, \ldots, m$. The following lemma is necessary to obtain $\Pr(R_{ef})$.

Lemma 11. *If $Q_1, Q_2, \ldots, Q_m$ are independent geometrically distributed random variables with success parameters $\lambda_1, \lambda_2, \ldots, \lambda_m$, and $T_m = \max(Q_1, Q_2, \ldots, Q_m)$, then the probability mass function on T_m is given as*

$$f_m(t) = \prod_{i=1}^{m}(1 - (1 - \lambda_i)^t) - \prod_{j=1}^{m}(1 - (1 - \lambda_i)^{t-1}). \tag{143}$$

Proof. The proof is omitted for the sake of brevity, but follows from induction on m. □

Armed with this lemma, we can obtain $\Pr(R_{ef})$ for the broadcast channel case.

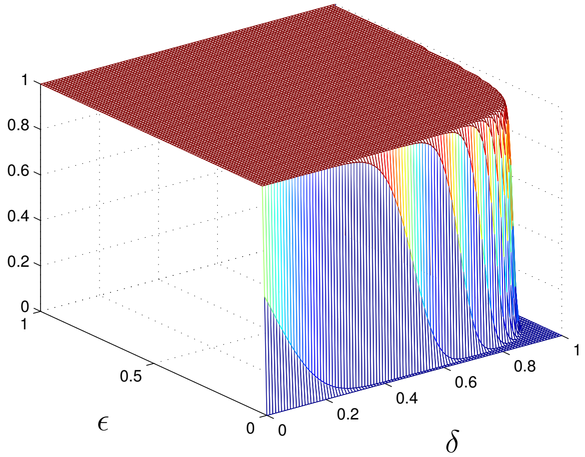

Figure 31: $\Pr(D \geq 50)$ when $\alpha = 1$ and $\eta = 5000$, as a function of the respective erasure probabilities in Q_m and Q_w, δ and ϵ.

Lemma 12. *Using the broadcast channel with m independent legitimate receivers and an eavesdropper*

$$\begin{aligned}\Pr(R_{ef}) &= \sum_{i=1}^{m}\left(\frac{1-\epsilon}{1-\epsilon\delta_i}\right) - \sum_{i<j}\left(\frac{1-\epsilon}{1-\epsilon\delta_i\delta_j}\right) + \sum_{i<j<k}\left(\frac{1-\epsilon}{1-\epsilon\delta_i\delta_j\delta_k}\right) - \cdots \\ &\quad +(-1)^{m+1}\left(\frac{1-\epsilon}{1-\prod_{i=1}^{m}\delta_i}\right) \qquad (144)\end{aligned}$$

where the notation $i < j$ means the summation traverses over all pairs (i, j) such that $i, j \in \{1, 2, \ldots, m\}$ and $i < j$, and similarly for $i < j < k$, etc.

Proof. Note that if the ith user requests a single packet until it is received, and in each transmission it is received with probability δ_i, then the total number of times the user must request the packet is governed by a geometric random variable with success parameter $1 - \delta_i$ [70]. Define $W_1, W_2, \ldots, W_m$ as the geometric random variables governing the total number of transmissions necessary for users $1, 2, \ldots, m$, respectively, to obtain the packet error free. Then, let $W = \max(W_1, W_2, \ldots, W_m)$. W governs the total number of transmissions necessary for all legitimate parties to receive the packet.

By Lemma 11, we know that

$$\Pr(W = w) = \prod_{i=1}^{m}(1 - \delta_i^w) - \prod_{j=1}^{m}(1 - \delta_i^{w-1}). \tag{145}$$

Finally, we point out that

$$\prod_{i=1}^{m}(1 - \delta_i) = 1 - \sum_{i=1}^{m} \delta_i + \sum_{i<j} \delta_i\delta_j - \sum_{i<j<k} \delta_i\delta_j\delta_k + \cdots (-1)^m \prod_{i=1}^{m} \delta_i. \tag{146}$$

Therefore,

$$\begin{aligned} \Pr(W = w) &= \sum_{i=1}^{m} \delta_i^{w-1}(1 - \delta_i) - \sum_{i<j}(\delta_i\delta_j)^{w-1}(1 - \delta_i\delta_j) + \cdots \\ &\quad +(-1)^{m+1}(\prod_{i=1}^{m} \delta_i)^{w-1}(1 - \prod_{i=1}^{m} \delta_i). \end{aligned} \tag{147}$$

With these pieces in place, we commence proving the lemma.

$$\begin{aligned} \Pr(R_{ef}) &= \sum_{w=1}^{\infty} \Pr(R_{ef}|W = w) \Pr(W = w) \\ &= \sum_{w=1}^{\infty}(1 - \epsilon^w) \left(\sum_{i=1}^{m} \delta_i^{w-1}(1 - \delta_i) - \sum_{i<j}(\delta_i\delta_j)^{w-1}(1 - \delta_i\delta_j) + \cdots \right. \\ &\quad \left. + (-1)^{m+1}(\prod_{i=1}^{m} \delta_i)^{w-1}(1 - \prod_{i=1}^{m} \delta_i) \right) \\ &= \sum_{i=1}^{m} \frac{1 - \delta_i}{\delta_i} \sum_{w=1}^{\infty}(1 - \epsilon^w)\delta_i^w - \sum_{i<j} \frac{1 - \delta_i\delta_j}{\delta_i\delta_j} \sum_{w=1}^{\infty}(1 - \epsilon^w)(\delta_i\delta_j)^w + \cdots \\ &\quad +(-1)^{m+1} \frac{1 - \prod_{i=1}^{m} \delta_i}{\prod_{i=1}^{m} \delta_i} \sum_{w=1}^{\infty}(1 - \epsilon^w)(\prod_{i=1}^{m} \delta_i)^w \\ &= \sum_{i=1}^{m} \frac{1 - \delta_i}{\delta_i} \left(\sum_{w=0}^{\infty} \delta_i^w - \sum_{w=0}^{\infty}(\epsilon\delta_i)^w \right) - \sum_{i<j}^{m} \frac{1 - \delta_i\delta_j}{\delta_i\delta_j} \left(\sum_{w=0}^{\infty}(\delta_i\delta_j)^w - \right. \\ &\quad \left. \sum_{w=0}^{\infty}(\epsilon\delta_i\delta_j)^w \right) + \cdots + (-1)^{m+1} \frac{1 - \prod_{i=1}^{m} \delta_i}{\prod_{i=1}^{m} \delta_i} \left(\sum_{w=0}^{\infty}(\prod_{i=1}^{m} \delta_i)^w - \sum_{w=0}^{\infty}(\epsilon \prod_{i=1}^{m} \delta_i)^w \right) \\ &= \sum_{i=1}^{m} \left(\frac{1 - \epsilon}{1 - \epsilon\delta_i} \right) - \sum_{i<j}^{m} \left(\frac{1 - \epsilon}{1 - \epsilon\delta_i\delta_j} \right) + \cdots + (-1)^{m+1} \left(\frac{1 - \epsilon}{1 - \prod_{i=1}^{m} \delta_i} \right). \end{aligned} \tag{148}$$

□

5.6.5 Collaborating Eavesdroppers

In this section we consider the case with l eavesdroppers working together to obtain the cryptogram $\mathbf{E}$, each with a possibly unique probability of packet erasure $\epsilon_1, \epsilon_2, \ldots, \epsilon_l$. All are assumed to obtain packets through independent PECs. It is simpler to first consider a single legitimate user Bob with probability of packet erasure δ. Then the general result that assumes m friendly parties with l collaborating eavesdroppers comes easily.

Lemma 13. *For l eavesdroppers and a single legitimate receiver,*

$$\Pr(R_{ef}) = \frac{1 - \prod_{i=1}^{l} \epsilon_i}{1 - \delta \prod_{i=1}^{l} \epsilon_i}. \tag{149}$$

Proof. The proof is straightforward if we note that collaborating eavesdroppers receive a single sent packet if at least one of them obtains the packet error free. Let W be a geometric random variable with success parameter $1 - \delta$. This governs the number of transmissions for each packet, as in the proof of Lemma 10. Then,

$$\begin{aligned}
\Pr(R_{ef}) &= \sum_{w=1}^{\infty} \Pr(R_{ef}|W = w) \Pr(W = w) \\
&= \sum_{w=1}^{\infty} (1 - (\prod_{i=1}^{l} \epsilon_i)^w)(1 - \delta)\delta^{w-1} \\
&= \frac{1 - \prod_{i=1}^{l} \epsilon_i}{1 - \delta \prod_{i=1}^{l} \epsilon_i}. \qquad (150)
\end{aligned}$$

□

This answer provides an easy bridge to an extremely general result.

Corollary 2. *For m intended parties and l eavesdroppers with similar notation as before,*

$$\Pr(R_{ef}) = (1 - \epsilon') \left(\sum_{i=1}^{m} \frac{1}{1 - \epsilon' \delta_i} - \sum_{i<j} \frac{1}{1 - \epsilon' \delta_i \delta_j} + \cdots + (-1)^{m+1} \frac{1}{1 - \epsilon' \prod_{i=1}^{m} \delta_i} \right), \tag{151}$$

where $\epsilon' = \prod_{i=1}^{l} \epsilon_i$.

Proof. This proof follows directly from the techniques used in the proofs of Lemmas 12 and 13. □

5.7 *Correlated Erasures in* Q_m *and* Q_w

Although the independence assumption on erasures occurring over Q_m and Q_w is fair in many cases; physical deployment of the receiver antennas, the availability of line-of-sight, and the presence or absence of scatterers at the transmitter and receivers [73, 74] may result in parallel channels between a transmitter and multiple receivers having correlated erasures. Therefore, we also address the effects of correlation between packet erasures at the intended receiver and packet erasures at the eavesdropper. The security analysis for this case necessitates a clear understanding of bounds on the correlation coefficient. This allows analysis of security enhancement assuming best and worst correlation conditions. In many cases security enhancement can still be obtained, even when the eavesdropper has a better channel than the legitimate receiver and erasures are correlated. In fact, correlation cannot reduce $\mathbb{E}[D]$ to zero if the legitimate receiver's channel quality is strictly better than that of the eavesdropper.

5.7.1 Pearson Correlation Coefficient

Since we still assume Q_m and Q_w to be memoryless, erasures occur independently with respect to other erasures in the same channel; however, erasures of the same packet but across different channels are correlated with correlation coefficient ρ. Let E_m and E_w be Bernoulli random variables that take on values in the set $\{0, 1\}$, where one signifies erasure and zero signifies error-free reception of a packet. Then, $\Pr(E_m = 1) = \delta$ and $\Pr(E_w = 1) = \epsilon$. The covariance of two random variables A and B is defined as [70]

$$\mathrm{cov}(A, B) = \mathbb{E}[(A - \mathbb{E}[A])(B - \mathbb{E}[B])], \tag{152}$$

and the variance of a random variable A can be expressed as

$$\mathrm{var}(A) = \mathrm{cov}(A, A). \tag{153}$$

Given these definitions, the Pearson correlation coefficient between random variables E_m and E_w is [70]

$$\begin{aligned}\rho &= \frac{\mathrm{cov}\left(E_m, E_w\right)}{\sqrt{\mathrm{var}\left(E_m\right)\mathrm{var}\left(E_w\right)}} \\ &= \frac{\mathbb{E}[E_m E_w] - \mathbb{E}[E_m]\mathbb{E}[E_w]}{\sqrt{(\mathbb{E}[E_m^2] - \mathbb{E}[E_m]^2)(\mathbb{E}[E_w^2] - \mathbb{E}[E_w]^2)}} \\ &= \frac{\mathbb{E}[E_m E_w] - \delta\epsilon}{\sqrt{\delta(1-\delta)\epsilon(1-\epsilon)}}. \end{aligned} \tag{154}$$

The last step is made using the first and second moments of a Bernoulli random variable, where $\mathbb{E}[E_m] = \mathbb{E}[E_m^2] = \delta$ and $\mathbb{E}[E_w] = \mathbb{E}[E_w^2] = \epsilon$. Let $p_{ij} = \Pr(E_m = i, E_w = j)$ [75]. Then, $\delta = p_{10} + p_{11} = \mathbb{E}[E_m]$ and $\epsilon = p_{01} + p_{11} = \mathbb{E}[E_w]$. It is trivial to show that $\mathbb{E}[E_m E_w]$ is equal to $\Pr(E_m = 1, E_w = 1)$. Thus, (154) can be expressed as

$$\rho = \frac{p_{11} - \delta\epsilon}{\sqrt{\delta(1-\delta)\epsilon(1-\epsilon)}}. \tag{155}$$

The Pearson correlation coefficient is commonly used to indicate the degree to which two random variables are similar. Although $|\rho| \leq 1$, it is a common misconception that ρ can take on any value from -1 to $+1$. In reality, there are bounds to the coefficient that are a function of the distribution of the random variables involved [76]. In our case, we have allowed δ and ϵ to take on any value in $[0, 1]$. We also know that $\delta = p_{11} + p_{10}$, $\epsilon = p_{11} + p_{01}$, and $p_{00} + p_{01} + p_{10} + p_{11} = 1$. Since $p_{ij} \geq 0$ for $i, j \in \{0, 1\}$, then

$$\max(\delta + \epsilon - 1, 0) \leq p_{11} \leq \min(\delta, \epsilon). \tag{156}$$

The bounds on p_{11} can be translated to bounds on ρ as

$$\frac{\max(\delta + \epsilon - 1, 0) - \delta\epsilon}{\sqrt{\delta\epsilon(1-\delta)(1-\epsilon)}} \leq \rho \leq \frac{\min(\delta, \epsilon) - \delta\epsilon}{\sqrt{\delta\epsilon(1-\delta)(1-\epsilon)}}. \tag{157}$$

For example, if $\delta = 0.3$ and $\epsilon = 0.15$, then $-0.275 \leq \rho \leq 0.642$.

5.7.2 Security Results for Correlated Erasures

In presenting results for the correlated case, we again assume that the degree distribution for the LDPC code and the puncturing pattern R are chosen such that $|R| = n' - k$. Thus, the MP decoder achieves the ML performance, and furthermore, D is equivalent in each decoded codeword.

The general results of Lemma 7 and Theorem 5 still hold for the correlated erasure case, because although erasures in Q_m and Q_w are correlated, each packet is received error free by Eve independent from other packets. Therefore, the reception of each transmitted packet can still be considered a Bernoulli experiment. The sum of the missing packets remains binomial with *success* parameter $1 - \Pr(R_{ef})$. Therefore, as with the previous generalizations of the security analysis, we only need to calculate $\Pr(R_{ef})$ under the new assumption, i.e., when erasure events over Q_m and Q_w are correlated.

Lemma 14. *In the packet erasure wiretap channel scenario with feedback for ARQ, and where Lemma 7 and Theorem 5 are satisfied, if channel erasures are correlated events across Q_m and Q_w with correlation coefficient ρ, then*

$$\Pr(R_{ef}) = \frac{1 - \epsilon}{1 - \epsilon\delta - \rho\sqrt{\delta\epsilon(1-\delta)(1-\epsilon)}}. \tag{158}$$

Proof. Let W be the total number of times that Bob requests a single packet over Q_m before he obtains it error free. Since Q_m is memoryless, the packet is erased each time independently with probability δ. Therefore, W is a random variable that takes on the number of total transmissions up to and including the first successful reception of the packet, and is thus geometrically distributed with success parameter $1 - \delta$. Then, [70]

$$\Pr(W = j) = (1 - \delta)\delta^{j-1}. \tag{159}$$

Let E_m^i and E_w^i denote the respective erasure outcomes in Q_m and Q_w for the ith retransmission of the packet, where a one signifies an erased packet as before. Therefore,

$$\begin{aligned}
\Pr\left(R_{ef}\right) &= \sum_{j=1}^{\infty} \Pr(R_{ef}|W=j)\Pr(W=j) \\
&= \sum_{j=1}^{\infty} (1-\Pr(E_w^1 = \cdots = E_w^j = 1 | E_m^1 = \cdots = E_m^{j-1} = 1, E_m^j = 0)\Pr(W=j) \\
&= \sum_{j=1}^{\infty} \left(1 - \left(\frac{p_{11}}{\delta}\right)^{j-1} \frac{p_{01}}{1-\delta}\right)(1-\delta)\delta^{j-1} \\
&= (1-\delta)\sum_{j=1}^{\infty} \delta^{j-1} - p_{01}p_{11}^{j-1} \\
&= 1 - \frac{p_{01}}{p_{11}}\left[\left(\sum_{j=0}^{\infty} p_{11}^j\right) - 1\right] \\
&= \frac{1-p_{11}-p_{01}}{1-p_{11}}. \qquad (160)
\end{aligned}$$

Now we calculate $p_{11} = \rho\sqrt{\delta\epsilon(1-\delta)(1-\epsilon)} + \delta\epsilon$ and $p_{01} = \rho\sqrt{\delta\epsilon(1-\delta)(1-\epsilon)} + \epsilon(1-\delta)$, to obtain

$$\Pr(R_{ef}) = \frac{1-\epsilon}{1-\epsilon\delta-\rho\sqrt{\delta\epsilon(1-\delta)(1-\epsilon)}}.$$

□

Clearly, this reduces to the independent case in (139) when $\rho = 0$. Figures 32 and 33 give examples of $\Pr(D \geq \beta)$ using this expression for $\Pr(R_{ef})$ where β is chosen to be one and 50, respectively. Both figures assume rate-1/2 LDPC codes in the encoder, and maximum-sized puncturing sets R. Figure 32 shows the result when the blocklength $n' = 200$, $|R| = 100$, and the packing factor $\alpha = 1$. Figure 33 assumes $n' = 10{,}000$, $|R| = n' - k = 5000$, and $\alpha = 1$. Therefore, $\eta = \frac{n}{\alpha} = 100$ for the first case, and $\eta = n/\alpha = 5000$ for the second case. We set $\delta = 0.5$ and plot different ϵ values as ρ takes on all possible values indicated by the bounds in (157). Both figures imply the existence of a correlation threshold ρ_{th} for $\Pr(D \geq \beta)$, in that if all other parameters are set, then for $\rho < \rho_{th}$, $\Pr(D \geq \beta)$ is essentially one, and for $\rho > \rho_{th}$, $\Pr(D \geq \beta)$ is essentially zero. The differences in the two figures show the result on security caused by increasing n', and thus increasing η. The main difference is the sharpness of the falloff in the curves, indicating that the threshold is better defined for larger η. Furthermore—and trivially—more degrees of freedom can be obtained with larger blocklength, even for higher correlation factors simply because there are more packets that could be lost to the eavesdropper. For instance, notice in Figure 33 that when $\epsilon = 0.51$, $\Pr(D \geq 50) \approx 1 \forall \rho$. There is no equivalent effect in Figure 32 for smaller blocklength.

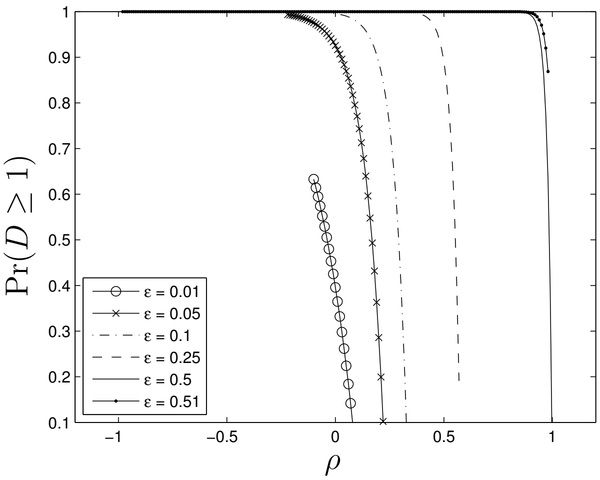

Figure 32: $\Pr(D \geq 1)$ when the number of packets $\eta = 100$, $\alpha = 1$, and Bob's erasure probability $\delta = 0.5$. Results are plotted for varying erasure probabilities ϵ for Eve's channel as a function of the correlation coefficient ρ.

Now that we understand how correlation affects D, we evaluate the extreme cases in correlation coefficients by considering the bounds on ρ in (157).

Consider the lower bound

$$\rho = \frac{\max(\delta + \epsilon - 1, 0) - \delta\epsilon}{\sqrt{\delta\epsilon(1-\delta)(1-\epsilon)}}. \tag{161}$$

Then,

$$\Pr(R_{ef}) = \frac{1-\epsilon}{1 - \max(\delta + \epsilon - 1, 0)}. \tag{162}$$

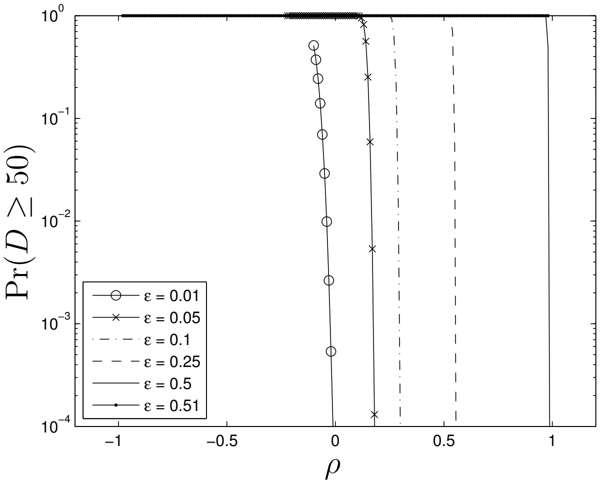

Figure 33: $\Pr(D \geq 50)$ when the number of packets $\eta = 5000$, $\alpha = 1$, and Bob's erasure probability $\delta = 0.5$. Results are plotted for varying erasure probabilities ϵ for Eve's channel as a function of the correlation coefficient ρ.

Therefore,

$$\Pr(R_{ef}) = \begin{cases} \frac{1-\epsilon}{2-\delta-\epsilon} & \text{if } \delta + \epsilon > 1 \\ \\ 1 - \epsilon & \text{otherwise} \end{cases} \tag{163}$$

If $\delta + \epsilon > 1$, this implies that $\Pr(R_{ef}) > 1 - \epsilon$. When $\delta + \epsilon \leq 1$, $\Pr(R_{ef}) = 1 - \epsilon$ implies that negative correlation can reduce the eavesdropper to an effective erasure channel where only one chance is given to intercept each packet, despite retransmission of some packets. Of course, the reasoning behind this is that this minimum correlation indicates that all of Eve's missing packets are obtained by Bob in the first transmission with probability one.

Now consider the upper bound

$$\rho = \frac{\min(\delta, \epsilon) - \delta\epsilon}{\sqrt{\delta\epsilon(1-\delta)(1-\epsilon)}}. \tag{164}$$

Then,

$$\Pr(R_{ef}) = \frac{1-\epsilon}{1 - \min(\delta, \epsilon)}, \tag{165}$$

thus implying that when Eve has at least as good of a channel as Bob, i.e. $\delta \geq \epsilon$, that the upper bound yields perfect correlation, and every packet is eventually received by Eve error free. However, if Bob can maintain a channel advantage over Eve such that $\delta < \epsilon$, then we see that

$$\Pr(R_{ef}) = \frac{1-\epsilon}{1-\delta} < 1. \tag{166}$$

Thus, even maximum correlation cannot reduce $\mathbb{E}[D]$ to zero. Since $\mathbb{E}[D]$ grows with k in (119), this indicates that we can still gain as many degrees of freedom as we desire on average by increasing the dimension of the encoder.

5.8 Ciphertext Error Rates

In Section 5.4.1, we argued that the descrambling operation will provide an expected error rate of 0.5 in the ciphertext output when an incorrect guess is made on the bits associated with the degrees of freedom. Here we provide additional discussion and simulation results that verify this argument. Let us explicitly define

$$\hat{\mathbf{P}}^{nL} = (\hat{\mathbf{P}}_1^n, \hat{\mathbf{P}}_2^n, \ldots, \hat{\mathbf{P}}_L^n) \tag{167}$$

to be the collection of punctured codewords obtained by Eve, and let

$$\hat{\mathbf{B}}^{n'L} = (\hat{\mathbf{B}}_1^{n'}, \hat{\mathbf{B}}_2^{n'}, \ldots, \hat{\mathbf{B}}_L^{n'}) \tag{168}$$

be the decoded codewords. Finally, define the block structure of Eve's decoder output as

$$\hat{\mathbf{E}}^{kL} = (\hat{\mathbf{E}}_1^k, \hat{\mathbf{E}}_2^k, \ldots, \hat{\mathbf{E}}_L^k). \tag{169}$$

Each channel-erased bit in the block $\hat{\mathbf{P}}_i$ yields a degree of freedom in $\hat{\mathbf{B}}_i$, and complete recovery of $\hat{\mathbf{B}}_i$ requires that D bits in $\hat{\mathbf{P}}_i$ be guessed correctly. If a guess is incorrect, then the decoding operation will still provide a valid codeword, and therefore, there will be at least as many errors in $\hat{\mathbf{B}}_i$ as the minimum distance of the LDPC code, i.e., the minimum Hamming distance between codewords. The descrambling process in (112) magnifies any errors in $\hat{\mathbf{B}}_i$ to an expected BER of 0.5 in $\hat{\mathbf{E}}_i$. Corollary 1 from Section 5.4.4 shows that for our encoder, it is possible to ensure the same for each block in $\hat{\mathbf{E}}$. Therefore, a brute-force attack on D bits must be accomplished to obtain each $\hat{\mathbf{E}}_i$.

Simulations of the end-to-end encoder and decoder were performed using the irregular LDPC code of Example 4 from Section 5.4.3 with $n' = 1000$, $k = 500$, and $L = 1$. Puncturing patterns used were such that $|R| \geq 498$ bits. Let γ be the number of bits in Eve's guess that are incorrect. Results are shown when $\gamma = 1$, 2, 3, 4, 5, 10, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, and 400 in Figure 34. Each γ value was tested 300 times on both the MP and ML decoder, while a new puncturing pattern R was generated every 10 experiments, and a new code from the ensemble was selected every 30 experiments. All tests produced BERs in $\hat{\mathbf{E}}$ between 0.414 and 0.578 with a mean of 0.5002. There is no noticeable difference between MP and ML decoders, or between γ values, as Figure 34 indicates.

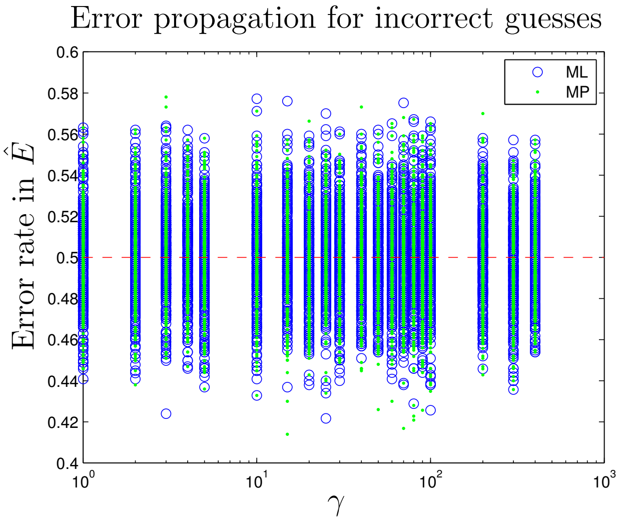

Figure 34: The simulated error rates in Eve's decoded cryptogram $\hat{\mathbf{E}}$ when γ errors are made in guessing bit values for D degrees of freedom in one of Eve's received codewords.

These results verify that unless D bits are guessed correctly for each codeword, Eve would be forced to attack the cryptographic layer of this system with an average BER of 0.5 in $\hat{\mathbf{E}}$. We can expect such an attack to fail miserably, as the indication is that the ciphertext could have been made just as reliable by flipping a coin for the bit values. As an example, the stream cipher analysis in Chapter 4 required much smaller error rates in the ciphertext to render both studied attacks completely ineffective.

5.9 Discussion and Conclusions

In this chapter, we presented and analyzed a practical physical-layer coding scheme that provides cryptographic security enhancements using channel coding techniques with ARQ. The design requirements are few because the system works for nearly every combination of channel parameters in the packet erasure wiretap channel. The security analysis reveals that this encoder provides degrees of freedom in an attacker's knowledge of the codewords. Since the attacker has no information about the bits associated with the degrees of freedom, a brute-force attack on these bits must be performed. The system propagates errors to an expected bit-error rate of 0.5 in the ciphertext for all guesses that are not exactly correct, and thus, the end result of the expected increase in attack complexity on the cryptosystem from our scheme is a multiplier that is exponential in the number of degrees of freedom. The system provides cryptographic security enhancement, even when eavesdroppers have an advantage over legitimate receivers in signal quality. The system was shown to still provide security, although a lesser amount, when the number of legitimate receivers or the number of collaborating attackers increases, or when erasures across the main channel and the wiretap channel are correlated.

CHAPTER VI

COMBINING PHYSICAL-LAYER SECURITY WITH CRYPTOGRAPHY

The results from both the channel coding design problem, and the cryptanalysis of error-prone ciphertext problem combine to make a compelling argument in favor of multilayer security. For one, a largely untapped source of security—the physical layer—is exploited to strengthen cryptography; and for two, in the absence of security from the physical layer, the cryptographic layer can still stand alone with no penalty on secrecy. If cryptography is the only source of security in a system, then channel codes will provide reliable ciphertext for friendly parties and eavesdroppers alike; thus, squandering the possible increase in security attainable from the physical layer. On the other hand, if physical-layer secrecy codes are the only source of security in a system, then there are many scenarios where eavesdroppers are likely to obtain information about the message, such as when an eavesdropper has better channel quality than the legitimate receiver. Therefore, a multilayer approach to security clearly outperforms standalone cryptography, and casts physical-layer security coding into a reasonable security role, that of security enhancement.

6.1 Conclusions

The objective of this research was to provide a multilayer security solution for transmitted data in a digital communications system using the combination of physical-layer security coding and application-layer cryptography. With this goal in mind, we first performed cryptanalysis with error-prone ciphertext for a simple substitution cipher in Chapter 3, and LFSR-based stream ciphers in Chapter 4. We concluded that a simple substitution cipher, although insecure as a standalone cipher nowadays, is often an ingredient in more complicated cryptosystems. Hence, the information-theoretic analysis we provided may yet have further applications in more modern ciphers. The analysis of stream ciphers differed from that of substitution ciphers, in that we looked at specific attacks and analyzed the utility of those attacks when error rates in the ciphertext were greater than zero. For stream ciphers, it was shown that small error rates in the ciphertext can render attacks useless. The attacks that were studied were of the fast-correlation variety, and positive error rates in the ciphertext tend to decorrelate data until the attacks can no longer extract information about the secret key. These results are indeed encouraging when we wish to enhance cryptography be providing error rates in ciphertext.

The second problem we addressed with regards to the multilayer security theme, focused on the design principles of practical channel coding techniques that exploit characteristics of the packet erasure wiretap channel model. We invented the family of stopping set codes, and analyzed their properties. They require feedback to guarantee reliable transmission for legitimate parties, and to obtain an effective advantage over eavesdroppers, even when they have lower erasure rates than legitimate receivers. These codes obscure transmitted ciphertext at an eavesdropper's receiver, and require brute-force attacks on a subset of missing bits that are associated with degrees of freedom in punctured codewords obtained by an eavesdropper.

Solutions to these two problems combine to form a more complete security solution than either solitary cryptography or isolated physical-layer security channel coding. Physical-layer security is cast into a cryptographic security enhancement role, reducing information-theoretic requirements of other less practical schemes to those that can be obtained in practice. While backing away from the fundamental information-theoretic security limit of secrecy capacity, design requirements are simplified to the point that the codes can be deployed in more uncertain scenarios, e.g. when eavesdroppers are undetected or channel state information is unknown. Furthermore, if an eavesdropper enjoys an advantage over legitimate parties, cryptography can still provide secrecy, even when physical-layer security is impossible. Channel codes that confuse an eavesdropper's ciphertext naturally force another layer of complexity on cryptographic attacks. Cryptanalysis techniques that take into account noisy ciphertext can allow this security enhancement to be quantified.

6.2 Future Work

The work presented here can be listed with many other contributions in the area of physical-layer security, although this work certainly leans toward the practical implementation aspects of the research area. However, our work on the cryptanalysis of noisy ciphertext, according to our knowledge, is the first of its kind, and may yet initiate more research in multilayer security. We also hope that this work will allow physical-layer security research to find better coding techniques that can apply to a multitude of different channel scenarios, and thus, become less dependent on perfect channel state information and other limiting requirements for real-world use.

REFERENCES

[1] J. Wales, "Wikipedia," http://www.wikipedia.org/, Apr. 2012.

[2] R. A. Mollin, *Codes The Guide to Secrecy from Ancient to Modern Times*, ser. Discrete Mathematics and Its Applications, K. H. Rosen, Ed. Boca Raton, FL: Chapman & Hall/CRC Taylor & Francis Group, 2005.

[3] A. D. Wyner, "The wire-tap channel," *Bell Syst. Tech. J.*, vol. 54, no. 8, pp. 1355–1387, Oct. 1975.

[4] M. Bloch and J. Barros, *Physical-Layer Security: From Information Theory to Security Engineering.* To Appear: Cambridge University Press, Sept. 2011.

[5] W. K. Harrison and S. W. McLaughlin, "Combining wiretap codes with the simple substitution cipher," in *Review at IEEE Information Theory Workshop*, Apr. 2012, pp. 1–5.

[6] ——, "Physical-layer security: Combining error control coding and cryptography," in *Proc. IEEE Int. Conf. Communications (ICC)*, Dresden, Germany, June 2009, pp. 1–5.

[7] ——, "Tandem coding and cryptography on wiretap channels: EXIT chart analysis," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Seoul, Korea, June-July 2009, pp. 1939–1943.

[8] ——, "EXIT charts applied to tandem coding and cryptography in a wiretap scenario," in *Proc. IEEE Information Theory Workshop*, Taormina, Sicily, Oct. 2009, pp. 173–177.

[9] W. K. Harrison, J. Almeida, D. Klinc, S. W. McLaughlin, and J. Barros, "Stopping sets for physical-layer security," in *Proc. IEEE Information Theory Workshop (ITW)*, Dublin, Ireland, Aug.-Sept. 2010, pp. 1–5.

[10] W. K. Harrison, J. Almeida, S. W. McLaughlin, and J. Barros, "Coding for cryptographic security enhancement using stopping sets," *IEEE Trans. Inf. Forens. Security*, vol. 6, no. 3, pp. 575–584, Sept. 2011.

[11] ——, "Physical-layer security over correlated erasure channels," in *Proc. IEEE Int. Conf. Communications. (ICC), [Online]. Available at http://arxiv.org/abs/1102.3641*, Ottawa, Canada, June 2012, pp. 1–5.

[12] T. K. Moon, *Error Correction Coding: Mathematical Methods and Algorithms.* Hoboken, NJ: John Wiley & Sons, Inc., 2005.

[13] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* Hoboken, NJ: John Wiley & Sons, Inc., 2006.

[14] D. R. Stinson, *Cryptography Theory and Practice*, 3rd ed., ser. Discrete Mathematics and Its Applications, K. H. Rosen, Ed. Boca Raton, FL: Chapman & Hall/CRC Taylor & Francis Group, 2006.

[15] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, no. 4, pp. 656–715, 1949.

[16] M. Bloch, J. Barros, M. R. D. Rodrigues, and S. W. McLaughlin, "Wireless information-theoretic security," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2515–2534, June 2008.

[17] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 623–656, July, Oct. 1948.

[18] I. Csiszár and J. Körner, "Broadcast channels with confidential messages," *IEEE Trans. Inf. Theory*, vol. 24, no. 3, pp. 339–348, May 1978.

[19] U. Maurer and S. Wolf, "Information-theoretic key agreement: From weak to strong secrecy for free," in *Advances in Cryptology — EUROCRYPT 2000*, ser. Lecture Notes in Computer Science, B. Preneel, Ed., vol. 1807. Springer-Verlag, May 2000, pp. 351–368.

[20] M. Bellare, S. Tessaro, and A. Vardy, "A cryptographic treatment of the wiretap channel," *IACR Cryptology ePrint Archive*, p. 15, 2012.

[21] L. H. Ozarow and A. D. Wyner, "Wire-tap channel II," *Bell Syst. Tech. J.*, vol. 63, no. 10, pp. 2135–2157, Dec. 1984.

[22] V. K. Wei, "Generalized Hamming weights for linear codes," *IEEE Trans. Inf. Theory*, vol. 37, no. 5, pp. 1412–1418, Sept. 1991.

[23] R. G. Gallager, *Low-Density Parity-Check Codes.* Cambridge, MA: MIT Press, 1963.

[24] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.

[25] A. Thangaraj, S. Dihidar, A. R. Calderbank, S. W. McLaughlin, and J.-M. Merolla, "Applications of LDPC codes to the wiretap channel," *IEEE Trans. Inf. Theory*, vol. 53, no. 8, pp. 2933–2945, Aug. 2007.

[26] A. T. Suresh, A. Subramanian, A. Thangaraj, M. Bloch, and S. W. McLaughlin, "Strong secrecy for erasure wiretap channels," in *Proc. IEEE Information Theory Workshop (ITW)*, Dublin, Ireland, Aug.-Sept. 2010, pp. 1–5.

[27] A. Subramanian, A. Thangaraj, M. Bloch, and S. McLaughlin, "Strong secrecy on the binary erasure wiretap channel using large-girth LDPC codes," *IEEE Trans. Inf. Forens. Security*, vol. 6, no. 3, pp. 585–594, Sept. 2011.

[28] M. Bellare and S. Tessaro, "Polynomial-time, semantically-secure encryption achieving the secrecy capacity," *IACR Cryptology ePrint Archive*, p. 22, 2012.

[29] M. Bloch, R. Narasimha, and S. W. McLaughlin, "Network security for client-server architecture using wiretap codes," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 3, pp. 404–413, Sept. 2008.

[30] Y. Liang, H. V. Poor, and L. Ying, "Secrecy throughput of MANETs with malicious nodes," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Seoul, Korea, June-July 2009, pp. 1189–1193.

[31] M. Baldi, M. Bianchi, and F. Chiaraluce, "Non-systematic codes for physical-layer security," in *Proc. IEEE Information Theory Workshop (ITW)*, Dublin, Ireland, Aug.-Sept. 2010, pp. 1–5.

[32] D. Klinc, J. Ha, S. W. McLaughlin, J. Barros, and B.-J. Kwak, "LDPC codes for the Gaussian wiretap channel," *IEEE Trans. Inf. Forens. Security*, vol. 6, no. 3, pp. 532–540, Sept. 2011.

[33] D. Klinc, J. Ha, S. McLaughlin, J. Barros, and B.-J. Kwak, "LDPC codes for the Gaussian wiretap channel," in *Proc. IEEE Information Theory Workshop (ITW)*, Taormina, Sicily, Oct. 2009, pp. 95–99.

[34] D. Klinc, J. Ha, S. W. McLaughlin, J. Barros, and B.-J. Kwak, "LDPC codes for physical layer security," in *Proc. IEEE Global Telecommunications Conf. (GLOBECOM)*, Honolulu, HI, Nov. 2009, pp. 1–6.

[35] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, July 2009.

[36] E. Hof and S. Shamai, "Secrecy-achieving polar-coding for binary-input memoryless symmetric wire-tap channels," *Submitted to IEEE Trans. Inf. Theory, Available online at http://arxiv.org/PS_cache/arxiv/pdf/1005/1005.2759v2.pdf*, Aug. 2010.

[37] H. Mahdavifar and A. Vardy, "Achieving the secrecy capacity of wiretap channels using polar codes," *Submitted to IEEE Trans. Inf. Theory, Available online at http://arxiv.org/PS_cache/arxiv/pdf/1007/1007.3568v1.pdf*, July 2010.

[38] R. J. Blom, "Bounds on key equivocation for simple substitution ciphers," *IEEE Trans. Inf. Theory*, vol. 25, no. 1, pp. 8–18, Jan. 1979.

[39] J. G. Dunham, "Bounds on message equivocation for simple substitution ciphers," *IEEE Trans. Inf. Theory*, vol. 26, no. 5, pp. 522–527, Sept. 1980.

[40] A. Sgarro, "Error probabilities for simple substitution ciphers," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 190–198, Mar. 1983.

[41] A. Orlitsky, N. P. Santhanam, K. Viswanathan, and J. Zhang, "Limit results on pattern entropy," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 2954–2964, July 2006.

[42] T. Siegenthaler, "Correlation-immunity of nonlinear combining functions for cryptographic applications (corresp.)," *IEEE Trans. Inf. Theory*, vol. 30, no. 5, pp. 776–780, Sept. 1984.

[43] V. S. Pless, "Encryption schemes for computer confidentiality," *IEEE Trans. Comput.*, vol. C-26, no. 11, pp. 1133–1136, Nov. 1977.

[44] P. R. Geffe, "How to protect data with ciphers that are really hard to break," *Electronics*, vol. 46, no. 1, pp. 99–101, Jan. 1973.

[45] J. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Inf. Theory*, vol. 15, no. 1, pp. 122–127, Jan. 1969.

[46] W. Blaser and P. Heinzmann, "New cryptographic device with high security using public key distribution," *IEEE Student Papers*, p. 150, 1979-1980.

[47] T. Siegenthaler, "Decrypting a class of stream ciphers using ciphertext only," *IEEE Trans. Comput.*, vol. C-34, no. 1, pp. 81–85, Jan. 1985.

[48] W. Meier and O. Staffelbach, "Fast correlation attacks on certain stream ciphers," *Journal of Cryptology*, vol. 1, pp. 159–176, 1989.

[49] V. V. Chepyzhov and B. J. M. Smeets, "On a fast correlation attack on certain stream ciphers," in *EUROCRYPT*, 1991, pp. 176–185.

[50] T. Johansson and F. Jonsson, "Theoretical analysis of a correlation attack based on convolutional codes," *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 2173–2181, Aug. 2002.

[51] A. Blum, M. Furst, M. Kearns, and R. Lipton, "Cryptographic primitives based on hard learning problems," in *CRYPTO '93*, ser. Lecture Notes in Computer Science, vol. 773, 1994, pp. 278–291.

[52] A. Blum, A. Kalai, and H. Wasserman, "Noise-tolerant learning, the parity problem, and the statistical query model," *Journal of the ACM*, vol. 50, no. 4, pp. 506–519, July 2003.

[53] M. P. C. Fossorier and M. J. Mihaljević, "A novel algorithm for solving the LPN problem and its application to security evaluation of the HB protocol for RFID authentication," in *INDOCRYPT 2000*, ser. Lecture Notes in Computer Science, R. Barua and T. Lange, Eds., vol. 4329, 2006, pp. 48–62.

[54] S. ten Brink, “Convergence of iterative decoding,” *Electronics Letters*, vol. 35, no. 10, pp. 806–808, May 1999.

[55] ——, “Convergence behavior of iteratively decoded parallel concatenated codes,” *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.

[56] A. Ashikhmin, G. Kramer, and S. ten Brink, “Extrinsic information transfer functions: Model and erasure channel properties,” *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2657–2673, Nov. 2004.

[57] U. Maurer and S. Wolf, “Secret-key agreement over unauthenticated public channels—Part I: Definitions and a completeness result,” *IEEE Trans. Inf. Theory*, vol. 49, no. 4, pp. 822–831, Apr. 2003.

[58] T. Richardson and R. Urbanke, *Modern Coding Theory.* New York, NY: Cambridge University Press, 2008.

[59] D. Burshtein and G. Miller, “An efficient maximum-likelihood decoding of LDPC codes over the binary erasure channel,” *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2837–2844, Nov. 2004.

[60] T. J. Richardson and R. L. Urbanke, “Efficient encoding of low-density parity-check codes,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 638–656, Feb. 2001.

[61] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing.* Upper Saddle River, NJ 07458: Prentice-Hall, Inc., 2000.

[62] K.-M. Lee and H. Radha, “The design of the maximum-likelihood decoding algorithm of LDPC codes over BEC,” in *Proc. 41st Annu. Conf. Information Sciences and Systems*, Baltimore, MD, Mar. 2007, pp. 463–468.

[63] C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke, “Finite-length analysis of low-density parity-check codes on the binary erasure channel,” *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1570–1579, June 2002.

[64] E. Rosnes and O. Ytrehus, “An efficient algorithm to find all small-size stopping sets of low-density parity-check matrices,” *IEEE Trans. Inf. Theory*, vol. 55, no. 9, pp. 4167–4178, Sept. 2009.

[65] A. Orlitsky, K. Viswanathan, and J. Zhang, “Stopping set distribution of LDPC code ensembles,” *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 929–953, Mar. 2005.

[66] A. Alloum, J. J. Boutros, G. I. Shamir, and L. Wang, “Non-systematic LDPC codes via scrambling and splitting,” in *Proc. Allerton Conf.*, Monticello, IL, Sept. 2005, pp. 1879–1888.

[67] G. I. Shamir and J. J. Boutros, “Non-systematic low-density parity-check codes for nonuniform sources,” in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Adelaide, South Australia, Sept. 2005, pp. 1898–1902.

[68] R. Diestel, *Graduate Texts in Mathematics: Graph Theory*, 3rd ed. Berlin, Germany: Springer-Verlag Berlin Heidelberg, 2006.

[69] A. Konheim, “A queueing analysis of two ARQ protocols,” *IEEE Trans. Commun.*, vol. 28, no. 7, pp. 1004–1014, July 1980.

[70] G. Grimmett and D. Stirzaker, *Probability and Random Processes*, 3rd ed. Oxford, UK: Oxford University Press, 2001.

[71] D. E. Knuth, “Johann Faulhaber and sums of powers,” *Mathematics of Computation (American Mathematical Society)*, vol. 61, no. 203, pp. 277–294, July 1993.

[72] T. Cover, “Broadcast channels,” *IEEE Trans. Inf. Theory*, vol. 18, no. 1, pp. 2–14, Jan. 1972.

[73] W. C.-Y. Lee, “Effects on correlation between two mobile radio base-station antennas,” *IEEE Trans. Commun.*, vol. 21, no. 11, pp. 1214–1224, Nov. 1973.

[74] H. Jeon, N. Kim, M. Kim, H. Lee, and J. Ha, “Secrecy capacity over correlated ergodic fading channel,” in *Proc. IEEE Military Communications Conf. (MILCOM)*, San Diego, CA, Nov. 2008, pp. 1–7.

[75] S. Zhao, D. Tuninetti, R. Ansari, and D. Schonfeld, “Multiple description coding over correlated multipath erasure channels,” in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Las Vegas, NV, Mar.-Apr. 2008, pp. 2153–2156.

[76] W. J. Shih and W.-M. Huang, “Evaluating correlation with proper bounds,” *Biometrics*, vol. 48, no. 4, pp. 1207–1213, Dec. 1992.