CODING TECHNIQUES FOR INFORMATION-THEORETIC STRONG SECRECY ON WIRETAP CHANNELS

A Thesis
Presented to
The Academic Faculty

by

Arunkumar Subramanian

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

Georgia Institute of Technology
December 2011

CODING TECHNIQUES FOR INFORMATION-THEORETIC STRONG SECRECY ON WIRETAP CHANNELS

Approved by:

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

Professor Alexandra Boldyreva
School of Computer Science
Georgia Institute of Technology

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

Professor Edward Coyle
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Professor Xiaoli Ma
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Date Approved: August 24, 2011

ACKNOWLEDGEMENTS

I wish to express my deepest gratitude to my advisor, Prof. Steven McLaughlin, for his unconditional support and encouragement during my years as a doctoral student. Thanks to the flexibility he extended to me, I was able to sample a variety of problems before narrowing down to the topic in this dissertation. I also thank my committee members—Prof. Faramarz Fekri, Prof. Xiaoli Ma, Prof. Edward Coyle, and Prof. Alexandra Boldyreva—for devoting their valuable time for my dissertation.

My sincere thanks goes to my research collaborators—Prof. Andrew Thangaraj and Ananda Theertha of IIT Madras, and Prof. Matthieu Bloch of GT Lorraine. I had great pleasure working with Andrew, whose collaboration led me to a majority of the ideas in this dissertation. I thank him for introducing me to the LDPC-code approach to secrecy; I also thank him for introducing me to the areas of coding theory and information theory during the final year of my undergraduate studies in IIT Madras. Andrew's clarity of thought and keen intuition are something I will always appreciate. Matthieu has always been there to lend his expertise on a variety of topics in physical-layer security. I am also grateful to some of his thorough *proof*-reading and proof-reading services! The work on stopping sets was initiated by Ananda and Andrew, before I joined them. I appreciate Ananda's dogged efforts when we were deciphering a particularly hard-to-read paper on stopping sets.

I enjoyed the pleasant company of my friends and colleagues in the Centergy fifth floor, especially Ramanathan Palaniappan, Sriram Lakshmanan, Matthieu Bloch, Willie Harrison, Demijan Klinc. Outside of my work life, I made some really great friends during my stay in Georgia Tech. I always looked forward spending weekends with the Centennial crowd—Nischint, Ramanan, Roshan, Rishi, Shubha, Sandeep,

and Yash—that usually involved the intriguing card game called *literature*. I thank Rishi and Shubha for several invited and uninvited dinners at their place. Badri helped me choose a lot of the mathematics courses in GT, most of which eventually found their use in this dissertation; I also thank him for the numerous badminton practice sessions.

This work would not have been possible without the support of my parents, who have been constantly anxious about my graduation for the last few years. They can breathe a sigh of relief now! I'm also thankful to my sister, Krithika, my brother-in-law, Karthick, and my brother, Ramanan, who have been a source of encouragement and helpful criticism during my PhD years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abbreviations

AWGN	additive white Gaussian noise.
BEC	binary erasure channel.
BER	bit-error rate.
BEWC	binary erasure wiretap channel.
BMC	binary-input memoryless channel.
BMSC	binary-input memoryless symmetric-output channel.
BP	belief propagation.
DDP	degree distribution pair.
DMC	discrete memoryless channel.
i.i.d.	independent and identically distributed.
LCM	least common multiple.
LDPC	low-density parity-check.
LLR	log-likelihood ratio.
LPS	Lubotzky-Phillips-Sarnak.
MAP	maximum a posteriori.
pdf	probability density function.

Mathematical Notation

x	A scalar (lowercase).
$\mathbf{x}$	A row vector (lowercase bold).
$\mathbf{x}^T$	A column vector.
$\mathbf{G}$	A matrix (uppercase bold).
X	A random variable (uppercase).
$\Pr(E)$	The probability of an event E.
$\mathrm{H}(X)$	The Shannon entropy of a discrete random variable X (in bits).
$\mathrm{I}(X;Y)$	The mutual information between the random variables X and Y.
$\mathbb{N}$	The set of all natural numbers, excluding zero.
$\mathbb{Q}$	The set of all rational numbers.
$\mathbb{R}$	The set of all real numbers.
$\mathbb{F}_q$	The finite field of size q.
$[n]$	The set of natural numbers up to n.

SUMMARY

Traditional solutions to information security in communication systems act in the application layer and are oblivious to the effects in the physical layer. Physical-layer security methods, of which information-theoretic security is a special case, try to extract security from the random effects in the physical layer. The wiretap channel model, where the transmitted symbols can be observed by a legitimate receiver and an eavesdropper through two different noisy channels, is of special interest in information-theoretic security. In information-theoretic security, there are two asymptotic notions of secrecy—weak and strong secrecy.

This dissertation investigates the problem of information-theoretic strong secrecy on the binary erasure wiretap channel (BEWC) with a specific focus on designing practical codes. The codes designed in this work are based on analysis and techniques from error-correcting codes. In particular, the dual codes of certain low-density parity-check (LDPC) codes are shown to achieve strong secrecy in a coset coding scheme.

First, we analyze the asymptotic block-error rate of short-cycle-free LDPC codes when they are transmitted over a binary erasure channel (BEC) and decoded using the belief propagation (BP) decoder. Under certain conditions, we show that the asymptotic block-error rate falls according to an inverse square law in block length, which is shown to be a sufficient condition for the dual codes to achieve strong secrecy.

Next, we construct large-girth LDPC codes using algorithms from graph theory and show that the asymptotic bit-error rate of these codes follow a sub-exponential decay as the block length increases, which is a sufficient condition for strong secrecy. The secrecy rates achieved by the duals of large-girth LDPC codes is shown to be an improvement over that of the duals of short-cycle-free LDPC codes.

CHAPTER I

INTRODUCTION

Communication systems and wireless networks have been growing very rapidly in the last few decades. The very nature of wireless systems allows anybody with a radio receiver to capture signals which may carry sensitive information. In addition, there is a growing trend among individuals and corporations to store personal and financial information on the Internet, potentially making it accessible to anybody with an Internet connection. Therefore, steps need to be taken to make sure that sensitive information may not be intercepted by malicious parties.

The traditional means of securing information is through cryptography, which is usually employed at the application layer; this security method is blind to what is happening at the physical layer. Recently, there has been some research on *physical-layer security* to develop techniques that take advantage of the properties of the physical communication medium to achieve security. There are several approaches to physical-layer security, and one of them is *information-theoretic* security. Cryptographic security assumes that the eavesdropper of a secure transmission has unlimited access to the transmission, but a limited processing power; due to this, conventional cryptography is sometimes also called *computational security*. On the other hand, information-theoretic security assumes that the eavesdropper has a limited access to the transmission, but an unlimited power to process it. This aspect of information-theoretic security is crucial because computation hardware is getting drastically cheaper every day; this means that computational security schemes that are currently considered secure will no longer be secure in the future.

For example, the Data Encryption Standard (DES) encryption scheme employs a

56-bit key and was approved as a standard by the U.S. National Bureau of Standards in 1976. In 1997, a DES cryptogram was broken for the first time in public. In 1998, the Deep Crack hardware broke a DES key in 56 hours [2]. The time taken to crack a DES key was reduced to less than 23 hours in 1999 using distributed computing. As computers keep getting more powerful every day, encryption schemes must also catch up with them. In this aspect, information-theoretic security is superior because it always assumes that the attacker has unbounded computing resources.

1.1 Physical-Layer Security

The point-to-point communication system with a single adversary is a fundamental model for studying information security. In this model, the transmitter sends a secure message to the legitimate receiver in such a way that the adversary cannot understand it. The legitimate parties are aware of the presence of the adversary and its abilities. In any secure communication scheme on the point-to-point communication system, the legitimate receiver must have some advantage over the adversary that can be used to achieve security. In the case of symmetric-key cryptography, the legitimate receiver knows the secret key used by the transmitter, which is unknown to the adversary. In asymmetric-key cryptography, the legitimate receiver can convey its public key, which can be verified by the transmitter to be authentic. This is a clear advantage over the adversary, who cannot mimic the actions of the legitimate receiver.

Any communication system that is deployed in the real world is subject to the random nature of physical communication media. These random effects include thermal (electronic) noise, interference, fading, etc. Suppose the point-to-point communication system is such that the physical communication medium affects the adversary's signal more unfavorably than the legitimate receiver's signal. In terms of signal quality, the legitimate receiver has a clear advantage over the adversary. Exploiting this

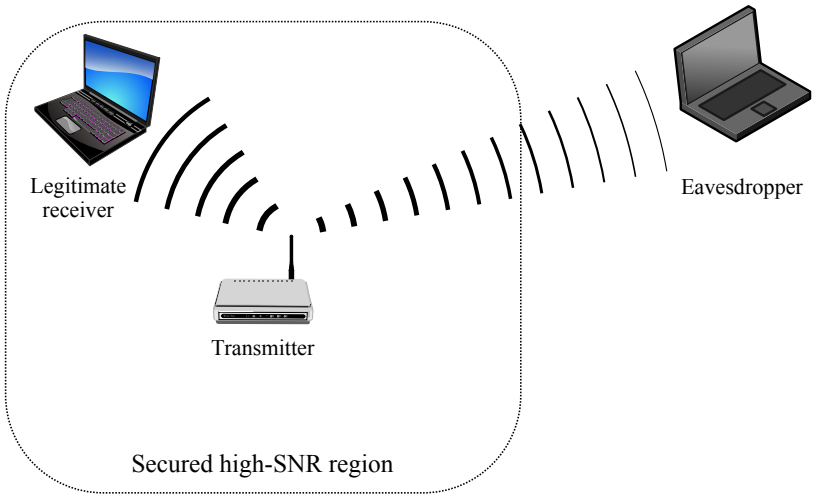


Figure 1: A wireless communication system with an eavesdropper outside the secured region.

advantage to achieve secure communication between the transmitter and the legitimate receiver is the primary objective of physical-layer security.

1.1.1 Some Motivating Examples

1. Consider a wireless system (Fig. 1) where a transmitter and a legitimate receiver are located in a secured area, say inside a building, that is inaccessible to an eavesdropper. The eavesdropper can only listen in on the wireless signal from outside the secured area, and therefore receives a signal with a lower intensity than the legitimate receiver.

2. Consider a distributed data-storage system where sensitive information must be stored in multiple storage nodes at different geographic locations. Once the data is stored, the system provides on-line access to a legitimate party to remotely read all the storage nodes. An illegitimate party (eavesdropper), who does not have on-line access to the nodes, can obtain offline access to some of the storage nodes. For example, the eavesdropper can bribe the caretaker of a particular storage node to get access to it. Assuming that the eavesdropper can

corrupt only a fraction of storage nodes in this manner, the legitimate party has a physical-layer advantage over the eavesdropper.

Though there are practical situations which can benefit from physical-layer security, the information-theoretic aspects of it are still not well-understood. Understanding the information-theoretic security aspects of simple abstract systems is an important stepping stone for designing information-theoretically secure practical systems. In this dissertation, we focus on theoretical scenarios involving abstract system models.

1.2 Dissertation Outline

This dissertation is organized as follows. In Chapter 2, we discuss some of the fundamental ideas in information-theoretic security and give a brief overview of some of the prior work related to this dissertation. In Chapter 3, we will introduce our channel model and our strong secrecy goal. We will also show in Chapter 3 that we can solve our strong secrecy problem by solving a related channel-coding problem. In the next two chapters, we will discuss two of our solutions to the related channel coding problem. Chapter 4 discusses our solution involving short-cycle-free low-density parity-check (LDPC) codes, where we show that LDPC codes without cycles of length two or four, and minimum left degree more than two can be used to construct a coding scheme that achieves strong secrecy on our channel model. In Chapter 5, we show that LDPC codes with girth growing logarithmically in block length can be used to achieve strong secrecy. Finally, we summarize our contributions and outline some future directions in Chapter 6.

CHAPTER II

THE FUNDAMENTALS OF INFORMATION-THEORETIC SECURITY

In this chapter, we summarize the basic ideas in information-theoretic security and discuss works that are closely related to this dissertation. In the first section, we give a quick overview of the definitions in information theory with regard to discrete random variables. In the next section, we summarize the notions of information-theoretic security on the wiretap channel. In the later sections, we will discuss some of the prior works regarding wiretap channels and secrecy coding for wiretap channels.

2.1 Information Theory—Basic Definitions

In this dissertation, we will deal with only discrete random variables, i.e., random variables distributed over a finite alphabet. This section summarizes the definitions in information theory that will be used in the subsequent chapters. These definitions are narrow in scope since our random variables are all discrete. For a more general discussion, we refer the reader to the book by Cover and Thomas [3].

Definition 2.1 (Shannon Entropy)**.** Given a discrete random variable X taking values in a finite alphabet $\mathcal{X}$ according to the probability distribution $p_X(x)$, the Shannon entropy of X is defined as

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

where we set $0 \log_2 0$ to 0 (strictly speaking, the summation is over the values for which $p_X(x) > 0$). ▼

Intuitively, the Shannon entropy $\mathrm{H}(X)$ is a quantitative measure of the randomness in X. In this dissertation, we call Shannon entropy as just "entropy."

Definition 2.2 (Conditional Entropy)**.** Given two random variables X and Y taking values in finite alphabets $\mathcal{X}$ and $\mathcal{Y}$, respectively, the conditional entropy $\mathrm{H}(Y|X)$ is defined as

$$\mathrm{H}(Y|X) = \sum_{x \in \mathcal{X}} p_X(x) \mathrm{H}(Y|X = x)$$

where $\mathrm{H}(Y|X = x)$ is the entropy of Y calculated using the conditional probability distribution $p_{Y|X}(y|x)$ over $y \in \mathcal{Y}$. ▼

The conditional entropy $\mathrm{H}(Y|X)$ is the measure of the randomness in Y given the knowledge of X.

Definition 2.3 (Mutual Information)**.** The mutual information between two discrete random variables X and Y is defined as

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

▼

The mutual information $\mathrm{I}(X;Y)$, which is a symmetric function, is the amount of information obtained about X by observing Y (and vice versa).

2.2 Information-theoretic Secrecy

The focus of this dissertation is the wiretap channel model, a generalized version of which is depicted in Fig. 2. This system consists of three parties,

1. Alice—the transmitter
2. Bob—the legitimate receiver
3. Eve—the eavesdropper

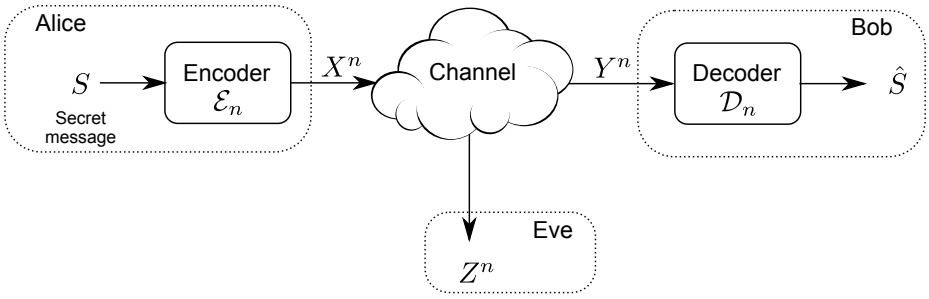


Figure 2: The generalized wiretap system model.

Eve is a "passive" eavesdropper in the sense that she cannot influence Alice or the channel in any way. This is one of the fundamental models used in classical cryptography and in information-theoretic security. It is important to note that the current works in classical cryptography consider fairly advanced attack models where the attacker can control the channel and also influence the transmitter. In fact, there are practical cryptographic schemes that provide security even in such models involving powerful adversaries. However, the area of information-theoretic security has not yet advanced to a sufficiently mature level to consider models involving powerful attackers.

Information-theoretic security usually considers the case where the wiretap channel is memoryless, and has discrete input alphabet and a discrete output alphabet. The input alphabet is $\mathcal{X}$, and the output alphabets are $\mathcal{Y}$ and $\mathcal{Z}$ for Bob and Eve, respectively. The alphabets $\mathcal{X}, \mathcal{Y}$ and $\mathcal{Z}$ are finite. For a memoryless channel, successive transmissions are independent of each other and the channel is defined by its joint transition probability $p_{YZ|X}(y, z|x)$.

Alice has a secret message S, uniformly distributed over a discrete alphabet $\mathcal{S}$, that she wants to convey to Bob through the wiretap channel in such a way that Eve cannot obtain S. To this end, Alice encodes S into a *cryptogram* X^n, which is an n-symbol vector over $\mathcal{X}$, and transmits it over the channel. In this case, the *block*

length of the encoder $\mathcal{E}_n$ is defined to be n and the rate $\mathtt{R}$ of the encoder is defined as

$$\mathtt{R} = \frac{\log_2 |\mathcal{S}|}{n}$$

The encoder and the channel characteristic $p_{YZ|X}$ are assumed to be publicly known, i.e., Eve's knowledge about the channel and the encoder is as good as Bob's. Alice's objective is to design the encoder such that the following objectives are met.

1. **Reliability:** Bob should be able to decode the secret S from his observation Y^n.

2. **Secrecy:** Eve should not be able to obtain S from her observation Z^n.

The precise definition of "secrecy" is the difference between cryptography and information-theoretic security. In cryptography, secrecy is defined in terms of the (infeasible) amount of computation required on Eve's part to decode S from Z^n. In the information-theoretic approach, secrecy is defined in terms of the mutual information between the secret and Eve's observation. Even under the information theory umbrella, there are different definitions of secrecy, namely, *Shannon perfect secrecy*, *strong secrecy*, and *weak secrecy*.

Definition 2.4 (Perfect Secrecy)**.** An encoder for the wiretap model achieves *perfect secrecy* in Shannon's sense if the probability of error in Bob's estimate $\hat{S}$ is zero and the mutual information between Eve's observation Z^n and the secret S is zero; that is,

$$\Pr(\hat{S} \neq S) = 0 \qquad \text{(Reliability)}$$

$$\mathrm{I}(S; Z^n) = 0 \qquad \text{(Secrecy)}$$

Equivalently, perfect secrecy is said to be achieved if the a priori probability distribution of S is the same as Eve's a posteriori probability distribution of S. ▼

2.2.1 Asymptotic Secrecy

Even when abstract wiretap models are considered, the above perfect secrecy criterion is achieved only in very few cases. The definition of perfect secrecy considers a single encoder $\mathcal{E}_n$ and a single secret message alphabet $\mathcal{S}$. In contrast, the notions weak and strong secrecy are asymptotic notions that consider a sequence of constant-rate encoders $(\mathcal{E}_{n_k})_{k\in\mathbb{N}}$ with increasing block lengths (n_k) and message alphabets $(\mathcal{S}_{n_k})$. For simplicity, the subscript k can be dropped. Since the rates of the encoders are all equal to $\mathtt{R}$, we have

$$\mathtt{R} = \frac{\log_2 |\mathcal{S}_n|}{n}, \qquad \forall n \in \{n_k\}$$

For a given encoder block length n, the asymptotic notions of secrecy consider the case where the message S, which is distributed uniformly over the message alphabet $\mathcal{S}_n$, is encoded using $\mathcal{E}_n$ into an n-symbol vector X^n and transmitted over the wiretap channel. The reliability criterion of the encoder sequence is specified in an asymptotic manner.

Definition 2.5 (Reliability)**.** An encoder sequence $(\mathcal{E}_n)$, with constant rate and increasing block length n, is said to achieve reliability on the wiretap channel if there exists a sequence of decoders $(\mathcal{D}_n)$ for which the probability of error in Bob's estimate $\hat{S} \triangleq \mathcal{D}_n(Y^n)$ of the secret S obeys

$$\lim_{n\to\infty} \Pr\left(S \neq \hat{S}\right) = 0$$

In other words, the probability that Bob decodes the message incorrectly goes to zero as n increases. ▼

Definition 2.6 (Weak Secrecy)**.** An encoder sequence $(\mathcal{E}_n)$, with constant rate and increasing block length n, is said to achieve *weak secrecy* on the wiretap channel if it achieves reliability, and the mutual information between Eve's observation Z^n and

the secret S is such that

$$\lim_{n\to\infty} \frac{1}{n} \mathrm{I}(S; Z^n) = 0$$

That is, the rate of information leakage to Eve converges to zero as n increases. ▼

Definition 2.7 (Strong Secrecy)**.** The encoder sequence $(\mathcal{E}_n)$, with constant rate and increasing block length n, is said to achieve *strong secrecy* on the wiretap channel if it achieves reliability, and the mutual information between Eve's observation Z^n and the secret S is such that

$$\lim_{n\to\infty} \mathrm{I}(S; Z^n) = 0$$

That is, the amount of information leaked to Eve goes to zero as n increases. ▼

A secret information rate $\mathtt{R}$ is said to be *achievable* with strong (weak) secrecy on the wiretap channel if there exists a sequence of encoders of rate $\mathtt{R}$ that achieve strong (weak) secrecy.

Definition 2.8 (Secrecy Capacity)**.** The strong secrecy capacity $\bar{C}_s$ of the wiretap channel is the supremum of all possible information rates $\mathtt{R}$ that are achievable with strong secrecy. The weak secrecy capacity C_s is defined as the supremum of all possible information rates that are achievable with weak secrecy. ▼

2.3 Historical Background

2.3.1 Perfect Secrecy using a Secret Key

The idea of using information theory to analyze cryptosystems was first introduced by Shannon in his 1949 paper. In [4], Shannon considers the system model illustrated in Fig. 3. Prior to transmission, Alice and Bob share a secret key K, which is a random variable distributed over a set of finite values. The key K is not revealed to Eve. The secret message S must be conveyed to Bob through a noiseless wiretap channel,

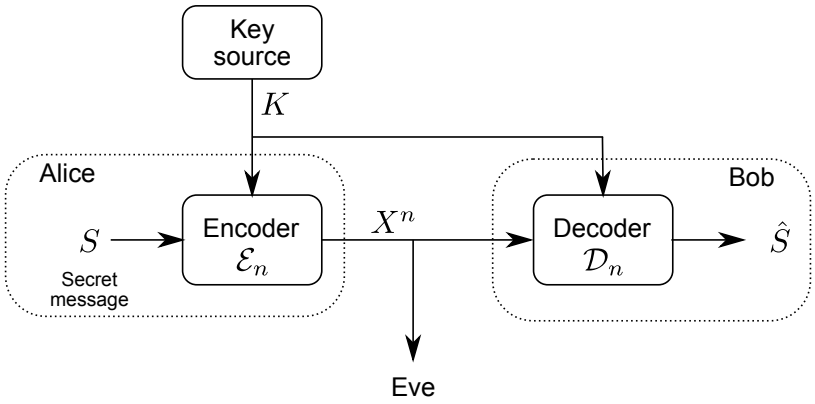


Figure 3: A schematic of Shannon's secrecy system

i.e., $X^n = Y^n = Z^n$. The encoder $\mathcal{E}_n$ uses both K and S as inputs and outputs the codeword X^n. Shannon showed that perfect secrecy on this model can be achieved only if

$$\mathrm{H}(K) \geq \mathrm{H}(S)$$

In other words, perfect secrecy can be achieved only if the secret key is at least as long as the secret message.

To achieve perfect secrecy on this model, Alice may encode S using Vernam's cipher, also known as the *one-time pad* [5]. For example, when S and K are independent uniformly distributed n-bit random vectors, the one-time pad encoder will output $X^n = S \oplus K$, where $\oplus$ denotes the bit-wise exclusive-OR operation. Given K, Bob can easily decode S from X^n. Eve, on the other hand, does not know K and the amount of information leaked to her by X^n is zero, i.e., $\mathrm{I}(S; X^n) = 0$.

However, we note that secure schemes that use a secret key require an additional channel to transmit the key. The channel for the secret key is assumed to be completely off-limits to the eavesdropper, and such a channel may not exist in certain scenarios. This additional requirement is not present most of the schemes in information-theoretic security, since they do not rely on a pre-shared secret key.

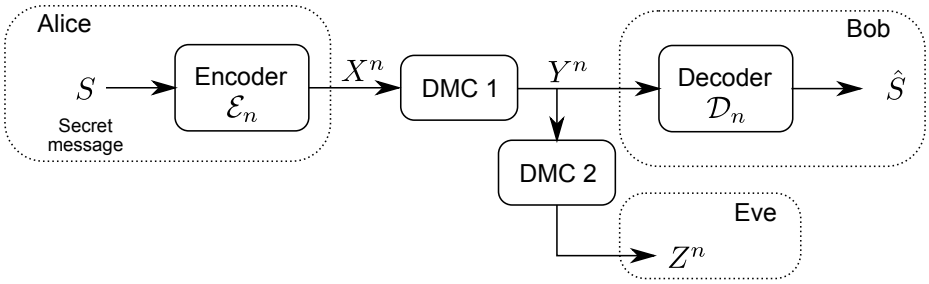


Figure 4: A schematic of Wyner's wiretap model.

2.3.2 Secrecy Capacity of Wiretap Channels

Degraded Wiretap Channels

The idea of analyzing cryptosystems with noise was introduced by Wyner in his 1975 paper on wiretap channels [6]. Wyner's wiretap model, illustrated in Fig. 4, consists of a discrete memoryless channels (DMCs) for both Bob and Eve, with Eve's channel being a degraded version of Bob's channel; in other words, $X^n \to Y^n \to Z^n$ forms a Markov chain. Wyner showed that whenever the channel capacity C_{M} of the main channel is more than the channel capacity C_{MW} of the wiretap channel (i.e., the capacity of the cascade of DMC 1 and DMC 2), the weak secrecy capacity C_s of the wiretap model is positive. In particular, he showed [6, Thm. 3] that

$$\begin{aligned} C_s &= \max_{p_X \in \mathcal{P}(C_s)} \left(\mathrm{I}(X;Y) - \mathrm{I}(X;Z)\right) \\ &\geq C_{\mathrm{M}} - C_{\mathrm{MW}} \end{aligned}$$

where $\mathcal{P}(C_s)$ represents the set of all distributions p_X on the single channel input X for which $\mathrm{I}(X;Y) \geq C_s$.

Non-degraded Wiretap Channels

Csiszár and Körner [7] generalized Wyner's model to cases where Eve's observation Z^n need not be a degraded version of Bob's observation Y^n. For their channel model, illustrated in Fig. 5, they showed [7, Cor. 2] that the weak secrecy capacity C_s is

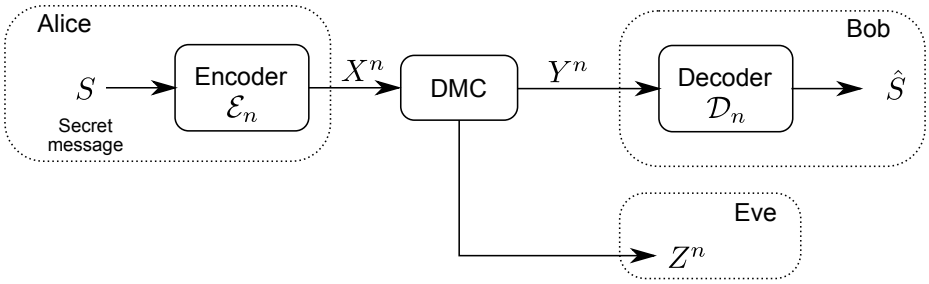


Figure 5: The wiretap model studied by Csiszár and Körner.

given by

$$C_s = \max_{V \to X \to YZ} \left(\mathrm{I}(V;Y) - \mathrm{I}(V;Z) \right)$$

where V is an auxiliary random variable such that $V \to X \to (Y,Z)$ forms a Markov chain and the maximization in the above expression is done over all such V. In [8], Körner and Marton introduced the notion of one channel being "less noisy" than another channel—channel 1, given by $X \to Y$, is said to be less noisy than channel 2, given by $X \to Z$, if

$$\mathrm{I}(V;Y) \geq \mathrm{I}(V;Z)$$

for all $V \to X \to (Y,Z)$. Under this notion, Csiszár and Körner proved [7, Thm. 3] that whenever the channel from Alice to Bob is less noisy than the channel from Alice to Eve, the weak secrecy capacity is given by

$$C_s = \max_{p_X} \left(\mathrm{I}(X;Y) - \mathrm{I}(X;Z) \right)$$

From Weak to Strong Secrecy

In the early years of the research on wiretap channels, the weak secrecy notion introduced by Wyner [6] was in predominant use. It can be seen that a scheme with weak secrecy can leak asymptotically unbounded amount of information, even though the leakage rate may be zero. For example, the amount of leaked information can scale

as $n^{1-\delta}$ for arbitrarily small $\delta > 0$; here the information leakage is unbounded while the leakage rate goes to zero. Due to this, Csiszár [9], and, independently, Maurer and Wolf [10] defined the notion of strong secrecy, and argued that this is a much better security condition compared to weak secrecy. In [9], it was proved that the technique of the earlier work by Csiszár and Körner [7], which focuses on achieving weak secrecy, actually leads to strong secrecy. In [10], Maurer and Wolf proposed a generic procedure to derive a strongly secret scheme from a weakly secret one using *extractors* [11, 12]. The key result of these two works is that for any wiretap channel, the weak secrecy capacity C_s is equal to its strong secrecy capacity $\bar{C}_s$.

2.4 Wiretap Codes

Until recently, the majority of the work on wiretap channels used non-constructive methods like the random-coding argument to show the existence of methods to achieve secrecy capacity. In particular, they did not provide explicit construction of encoders or *wiretap codes* to achieve secrecy, even at rates below the secrecy capacity. For example, Wyner [6, §V] uses a random-coding argument to prove his main result, while Csiszár and Körner [7, §IV] use an argument based on *typical sequences*.

Historically, the first *explicit* construction of wiretap codes was done in a slightly modified version of the wiretap channel with combinatorial constraints. For probabilistic channels, there are currently two approaches to secrecy coding—the LDPC code based approach, and the polar coding approach.

2.4.1 Wiretap Channel with Combinatorial Constraints

While the works of Wyner, and Csiszár and Körner are concerned with a discrete memoryless wiretap channel, the work by Ozarow and Wyner [13] considers a wiretap model with a combinatorial constraint. This channel, called a *type-II wiretap channel*, has a noiseless binary link between Alice and Bob such that a *constant fraction* of

the transmitted bits are leaked to Eve. That is, if n bits are transmitted over the channel and α is the leakage fraction, Eve is allowed to see $\mu = \alpha n$ of the n transmitted bits. Alice knows only the fraction α, but not the bit positions seen by Eve. Given the parameter α, the secrecy capacity of the type-II wiretap channel is $C_s = 1 - \alpha$. Ozarow and Wyner proved [13, Thms. 2 & 3] that it is possible to achieve the weak secrecy capacity of the type-II wiretap channel using a *coset coding scheme*. They also showed [13, Thm. 4] that there exists a coset coding based wiretap code that asymptotically leaks at most one bit to Eve; this result is stronger than the weak secrecy notion and only slightly weaker than the strong secrecy notion.

Unlike the probabilistic channels of [6,7], it is possible to achieve perfect secrecy on channels with combinatorial constraints. Wei [14] introduced the notion of *generalized Hamming weight* of linear codes and related it to the problem of achieving perfect secrecy on the type-II wiretap channel using coset coding.

Secret Sharing Schemes

In the cryptography community, a version of the type-II wiretap problem was studied under the name *secret sharing scheme*. The first constructions of secret sharing schemes were the ones by Blakley [15] and Shamir [16]. They studied the problem of distributing a secret among n parties such that a cooperation of any $k-1$ of them cannot recover the secret whereas coalitions of any k of them can. McEliece and Sarwate [17] showed that Shamir's scheme was equivalent to coset coding with a Reed-Solomon code.

In [18], Karnin, et al. generalized Shamir's threshold scheme to *ramp* secret sharing schemes. Here, a d-symbol long secret is broken into n different pieces so that any $k-d$ pieces or fewer do not give any information about the secret, whereas any k pieces give complete information about the secret. In the same work, it was also shown that any ramp scheme corresponds to a *maximum distance separable code* [19].

In a secret sharing scheme, the subsets of pieces that can recover the secret are called *access sets*, the subsets that convey no information about the secret are called *non-access sets*, and the remaining subsets are called *semi-access sets*. Under this terminology, Blakley and Shamir considered the case where all the subsets of size k are access sets and there are no semi-access sets; Karnin, et al. considered the case where sets of size k or more are access sets, sets of size $k - d$ or less are non-access sets and the remaining sets are semi-access sets.

The secret sharing problem was further generalized by Brickell and Davenport [20] and Kurosawa, et al. [21] to include access sets of more general structures. Under this framework, the secret sharing problem was linked [21] to combinatorial objects called *matroids* [22]. Massey [23, 24] studied secret sharing schemes based on puncturing linear codes and showed that the access structures in this scheme are related to the *minimal codewords* of the underlying linear code.

2.4.2 The LDPC Code Approach

The first explicit code construction for probabilistic channels was done by Thangaraj, et al. [1] using LDPC codes, where they consider a noiseless main channel and a binary erasure channel for the wiretapper. They use the coset coding scheme of [6, 13] using the duals of LDPC codes and show that this scheme achieves weak secrecy at rates close to secrecy capacity. The security mechanism of their technique is based on the *threshold effect* of LDPC codes—when the limiting performance of LDPC codes over a noisy channel is studied by varying the channel parameter, the asymptotic block-error rate suddenly drops from a non-zero value to zero. The approach in [1] was later extended to certain cases where the main channel is also noisy [25, 26].

2.4.3 The Polar Coding Approach

A new class of channel codes, called *polar codes*, were recently invented by Arıkan [27]. For a given binary input symmetric output channel, Arıkan showed that the polar codes designed for that channel achieve the channel capacity. The basic mechanism behind their capacity achieving property is *channel polarization*—by transforming n bits using a carefully chosen linear transformation and transmitting them over the channel, a portion of the bits ("bad" bits) can be made to appear as if they are transmitted through a pure-noise channel and the remaining bits ("good" bits) can be made to appear as if they are transmitted through a noiseless channel. The polar code is defined by assigning a preset value to the bad bits and transmitting the message using the good bits. It was shown in [27] that the fraction of the good bits approaches the capacity of the channel, which directly means that polar codes are capacity achieving.

The idea of designing wiretap codes using channel polarization was published by several research groups almost simultaneously, namely, by Mahdavifar and Vardy [28, 29], Andersson, et al. [30], Hof and Shamai [31], and Koyluoglu and El Gamal [32]. The polar coding technique for wiretap codes can be outlined as follows.

1. Transmit the secret message on bit positions that are good for Bob, but bad for Eve.

2. Transmit uniform independent and identically distributed (i.i.d.) random bits on bit positions that are good for both Bob and Eve.

3. Transmit zeros on bit positions that are bad for both Bob and Eve.

The above technique works because of the channel capacity achieving property of polar codes. The i.i.d. random bits are transmitted at a rate close to Eve's channel capacity and they can be perfectly decoded by Eve using a *successive cancellation* decoder [27]. This means that Eve's channel cannot carry any significant information

about the secret message since it would be a violation of Shannon's Noisy-Channel Coding Theorem [33].

In the aforementioned works on the polar coding technique, it was shown that this technique achieves the entire rate-equivocation region of degraded binary-input symmetric-output wiretap channels. In particular, their schemes achieve weak secrecy at rates close to the secrecy capacity. For the case of non-degraded wiretap channels and non-symmetric wiretap channels, this technique achieves weak secrecy, but at rates away from the secrecy capacity. Among these works, the technique of Mahdavifar and Vardy [29] is unique because it achieves strong secrecy for wiretap models with a noiseless main channel at rates arbitrarily close to secrecy capacity.

CHAPTER III

STRONG SECRECY OVER THE BINARY ERASURE WIRETAP CHANNEL

In this dissertation, we are concerned with the problem of designing encoders for achieving strong secrecy on the binary erasure wiretap channel (BEWC) model. In this chapter, we will introduce the BEWC model and do an information-theoretic analysis of the coset coding scheme for this model. We will then show (Lemma 3.3) that the duals of certain "good" channel codes for an appropriate binary erasure channel (BEC) can be used in a coset coding scheme to achieve strong secrecy on the BEWC.

3.1 System Model and Motivation

3.1.1 Binary Erasure Wiretap Channel Model

The BEWC model is a special case of the wiretap channel model introduced in Section 2.2. The BEWC model (Fig. 6), first studied by Thangaraj, et al. [1], consists of two legitimate parties, Alice and Bob, who want to communicate in the presence of a passive eavesdropper, Eve. The channel between Alice and Bob is a noiseless binary channel. The channel between Alice and Eve, denoted by BEC(ξ), is a BEC with erasure probability ξ. A bit sent through a BEC is either received unmodified or is erased. Whenever an erasure occurs, the channel outputs the erasure symbol '?'. The BEC is a memoryless channel, which means that bits sent successively are erased independently.

In the BEWC model, Alice's secret S is a random variable uniformly distributed over an alphabet $\mathcal{S}_n$. Alice's objective is to convey this secret message to Bob without

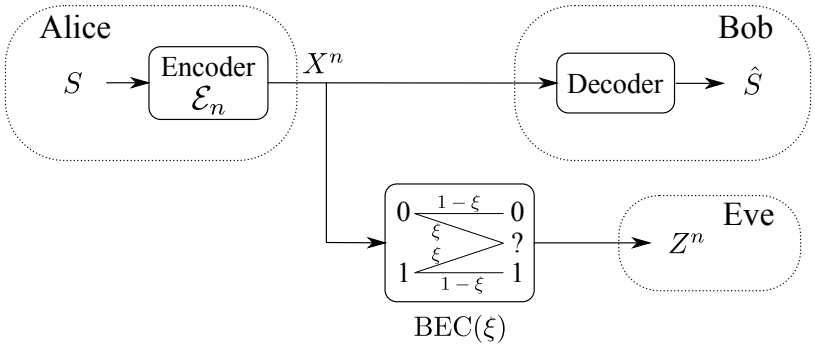


Figure 6: The binary erasure wiretap channel model.

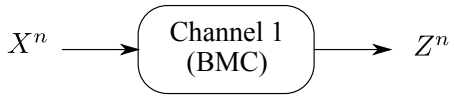
revealing it to Eve. In order to do this, Alice uses an encoder $\mathcal{E}_n$ to convert S into an n-bit random variable X^n and then transmits X^n over the BEWC. The rate of the encoder $\mathcal{E}_n$ is given by ratio of the input entropy to the output block length, i.e.,

$$\mathtt{R}_n = \frac{\mathrm{H}(S)}{n} = \frac{\log_2 |\mathcal{S}_n|}{n}$$

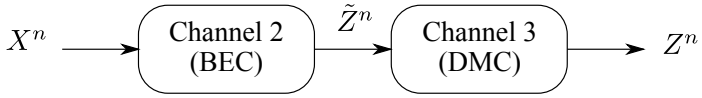
The encoder $\mathcal{E}_n$ is known both to Bob and Eve. In our research, we focus on achieving asymptotic secrecy and we consider a sequence $(\mathcal{S}_{n_k}, \mathcal{E}_{n_k})_{k\in\mathbb{N}}$ of message alphabet and encoder pairs such that the encoder rates are lower-bounded by $\mathtt{R} > 0$ and the block length n_k increases monotonically with k. The subscript k is present because the block length n need not increase in increments of one and its purpose is strictly to make the definition of the alphabet-encoder sequence precise. In order to make the notation easier, we will drop the subscript k and denote to the alphabet-encoder sequence using $(\mathcal{S}_n, \mathcal{E}_n)$.

3.1.2 Application of BEWC Analysis to Other Channel Models

In our work, we study the BEWC model because it is a fundamental model and its analysis is extendable to a lot of different wiretap models. For example, Rathi, et al. [25] consider the wiretap model where the main channel and the wiretap channel are independent BECs and their analysis is an extension of the BEWC analysis.



(a) A binary-input memoryless channel.



(b) An equivalent degraded binary erasure channel.

Figure 7: The decomposition of a BMC (a) into a degraded BEC (b).

Moreover, any wiretap model with a noiseless main channel and a binary-input memoryless channel (BMC) for the wiretapper can be modeled as a degraded BEWC. This is possible because of the Erasure Decomposition Lemma [34, Lemma 4.78] and its generalization [35, Prop. 6.4] to non-symmetric channels. According to this lemma, any BMC can be considered as the cascade of an appropriate BEC and an appropriate DMC.

For example, Channel 1 in Fig. 7, which is an arbitrary BMC, can be represented as the result of a cascade of Channel 2, a BEC, and Channel 3, a DMC. Consider the wiretap model where the main channel is noiseless and the wiretapper's channel is Channel 1 (Fig. 8a). To achieve strong (weak) secrecy on this wiretap model, we may use encoders that achieve strong (weak) secrecy on the BEWC model with Channel 2 as the wiretapper's channel (Fig. 8b). These encoders achieve $\mathrm{I}(S;\tilde{Z}^n) \to 0$ (or alternatively $\frac{1}{n}\mathrm{I}(S;\tilde{Z}^n) \to 0$ for weak secrecy). Since $X^n \to \tilde{Z}^n \to Z^n$ forms a Markov chain, we have the following well-known *data-processing inequality*.

$$\mathrm{I}(X^n;\tilde{Z}^n) \geq \mathrm{I}(X^n;Z^n) \tag{1}$$

From the above inequality, one can immediately see that codes that are designed to achieve strong (weak) secrecy on the BEWC model in Fig. 8b also achieve strong (weak) secrecy on the wiretap model in Fig. 8a.

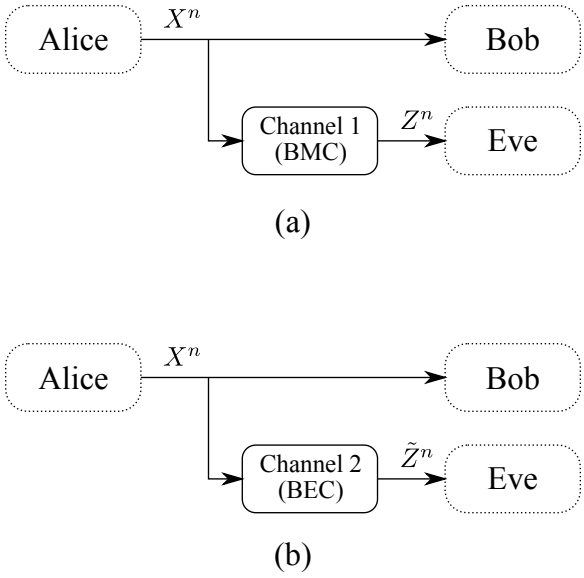


Figure 8: A wiretap model with a BMC wiretapper (a) and its corresponding BEWC model (b) based on the Erasure Decomposition Lemma.

It is important to note that the above approach to secrecy coding for BMC wiretap models cannot achieve secrecy capacity. In our example (Fig. 8), we see that $\mathrm{I}(X^n; Z^n)$ is bounded away from $\mathrm{I}(X^n; \tilde{Z}^n)$, except for trivial cases like when Channel 3 is noiseless. This means that the secrecy capacity C_s of the model in Fig. 8a, which is given by

$$C_s = \max_{P_X} \left(\mathrm{H}(X) - \mathrm{I}(X; Z)\right)$$

is strictly greater than that of the BEWC model in Fig. 8b, which is given by

$$\tilde{C}_s = \max_{P_X} \left(\mathrm{H}(X) - \mathrm{I}(X; \tilde{Z})\right)$$

Though the BEWC approach to achieving secrecy on BMC wiretaps is sub-optimal in terms of achievable rates, it is useful because we do not have to do any additional analysis of the BMC wiretap model.

3.1.3 Our Research Goal

The goal of our research is to design encoders $(\mathcal{E}_n)$ for Alice such that we are able to achieve strong secrecy on the BEWC model. Mathematically, the conditions for

strong secrecy are

1. There must exist a decoder for Bob such that estimate $\hat{S}$ of the secret message S must have a vanishing probability of (block) error. That is,

$$\Pr\left(\hat{S} \neq S\right) \to 0 \qquad \text{as } n \to \infty$$

2. The mutual information between Eve's observation Z^n and the secret message S must vanish as the block length increases. That is,

$$\mathrm{I}(S; Z^n) \to \infty \qquad \text{as } n \to \infty$$

3.2 The Coset Coding Scheme

The sequence of encoders $(\mathcal{E}_n)$ to be used in the BEWC model must have rate at least $\mathtt{R}$. In the classical channel or source coding scenario, the case where the $\mathcal{E}_n$'s are deterministic encoders or, equivalently, functions is usually considered. In contrast to deterministic encoders, *stochastic encoders* are not functions and have many possible outputs for a given input. In particular, given an input vector $\mathbf{s}$, a stochastic encoder $\mathcal{E}_n$ is such that its output $\mathcal{E}_n(\mathbf{s})$ is a random variable. In the secrecy coding scenario, deterministic encoders, in general, have a poorer secrecy performance compared to stochastic encoders. Due to this reason, almost all secrecy coding works deal only with stochastic encoders. In our work, we will be doing the same.

In the secrecy coding problem, Alice must not only keep the message S hidden from Eve; she must also reliably convey it to Bob. Therefore, we only consider stochastic encoders $\mathcal{E}_n$ for which the set of possible values of the random variables $\mathcal{E}_n(\mathbf{s})$ and $\mathcal{E}_n(\tilde{\mathbf{s}})$ are disjoint for $\mathbf{s} \neq \tilde{\mathbf{s}}$. In this case, the set of possible output vectors of the encoder $\mathcal{E}_n$ can be partitioned (Fig. 9) into different subsets such that all vectors in the a single subset correspond to only one possible input vector. Clearly, such an encoder will achieve reliability on the BEWC since there exists a straightforward

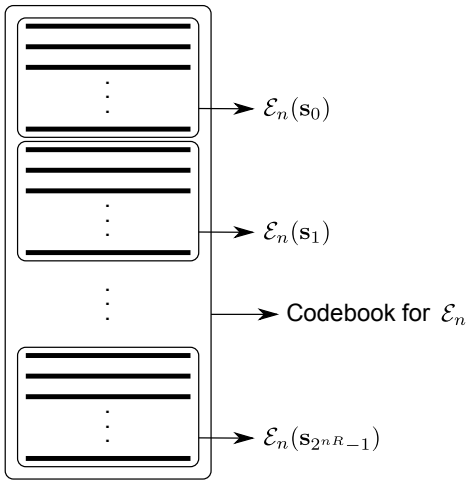


Figure 9: The partitioning of the set of all possible output vectors of a stochastic encoder $\mathcal{E}_n$ according to the input message. Each horizontal bold line represents an output codeword.

decoder with

$$\Pr\left(\hat{S} \neq S\right) = 0$$

3.2.1 Binary Linear Block Codes—Background

For integers n and k with $0 \leq k \leq n$, an (n, k) *binary linear block code* $\mathcal{C}$ is a k-dimensional vector subspace of the n-dimensional vector space $\mathbb{F}_2^n$ over the binary field $\mathbb{F}_2 \triangleq \{0, 1\}$. The number n is called the *length* or the *block length* of the code $\mathcal{C}$. The *rate* of the code $\mathcal{C}$ is defined as the ratio k/n. Due to its linearity, the code $\mathcal{C}$ can be defined as the row space of a carefully chosen $k \times n$ binary matrix $\mathbf{G}$. That is,

$$\mathcal{C} = \left\{\mathbf{mG} : \mathbf{m} \in \mathbb{F}_2^k\right\}$$

The matrix $\mathbf{G}$ is called a *generator matrix* of $\mathcal{C}$. The code $\mathcal{C}$ can also be defined as the null-space or kernel of an appropriate $(n-k) \times n$ binary matrix $\mathbf{H}$. That is,

$$\mathcal{C} = \left\{\mathbf{x} \in \mathbb{F}_2^n : \mathbf{Hx}^T = \mathbf{0}\right\}$$

In such a case, $\mathbf{H}$ is called a *parity-check matrix* of $\mathcal{C}$. Note that the parity-check matrix and the generator matrix are not necessarily unique for a given code.

3.2.2 Description

In our research, we will be using a class of stochastic encoding schemes called *coset coding schemes*. A coset coding scheme, which was introduced by Wyner [6], is based on a linear code and its cosets. Given the block length n and the rate $\mathtt{R}$ of the coset coding scheme, a binary linear block code $\mathcal{C}$ of block length n and rate $1-\mathtt{R}$ is used as a starting point. This code is a linear subspace of the n-dimensional vector space $\mathbb{F}_2^n$ over the binary field $\mathbb{F}_2 \triangleq \{0,1\}$. The code $\mathcal{C}$ has dimension $n(1-\mathtt{R})$ and therefore it has $2^{n\mathtt{R}}$ different cosets. Let $C_0, C_1, \ldots C_{2^{n\mathtt{R}}-1}$ be the cosets of $\mathcal{C}$. The input S to the coset coding scheme is an $n\mathtt{R}$-bit vector. Let $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{2^{n\mathtt{R}}-1}$ be the possible values for S (we will discuss how the possible messages are indexed in the next paragraph). Given $S = \mathbf{s}_i$, the coset coding scheme outputs a vector X^n, chosen uniformly at random from the coset C_i.

Let $\mathbf{H}$ be the parity-check matrix of the linear code $\mathcal{C}$. The matrix $\mathbf{H}$ is a $(n\mathtt{R})$-by-n binary matrix. Any n-bit vector $\mathbf{c}^n$ is a codeword in $\mathcal{C}$ if and only if it satisfies the system of parity-check equations given in the matrix form by

$$\mathbf{c}^n \ \mathbf{H}^T = 0$$

For any binary vector $\mathbf{x}^n$, the $n\mathtt{R}$-bit vector $\mathbf{x}^n\mathbf{H}^T$ is called the *syndrome* of $\mathbf{x}^n$ with respect to $\mathbf{H}$. It can be easily seen that two binary vectors $\mathbf{x}^n$ and $\tilde{\mathbf{x}}^n$ belong to the same coset of $\mathcal{C}$ if and only if they have the same syndrome. Based on this, cosets can be labeled according to their syndrome. For example, the coset corresponding to the input message $S = \mathbf{s}_i$ is $C_i = \{\mathbf{x}^n \in \mathbb{F}_2^n : \mathbf{x}^n \ \mathbf{H}^T = \mathbf{s}_i\}$.

3.2.3 Equivocation Analysis Under Erasures

Given the output vector X^n of the coset coding scheme, we can find its input S without ambiguity since the cosets $\{C_i\}$ are mutually disjoint. Suppose we receive a vector Z^n which is an erasure degraded version of X^n. The number and the position of these erasures may be random. Denote the set $\{1, 2, \ldots, n\}$ using $[n]$. Let J be the random vector containing the indices of the erased bits and let $\mathbf{j}$ be a particular instance of J. We have the following observation due to Ozarow and Wyner [13].

Lemma 3.1 ([13, Lemma 4]). *For an index set* $\mathbf{j} \subseteq [n]$ *,* $\mathrm{I}(S; Z^n | J = \mathbf{j}) = 0$ *if and only if*

$$\mathrm{rank}(\mathbf{G}_{[n]\setminus\mathbf{j}}) = n - |\mathbf{j}|$$

Here, $\mathbf{G}_{[n]\setminus\mathbf{j}}$ denotes the sub-matrix of $\mathbf{G}$ constructed using the columns indexed by $[n] \setminus \mathbf{j}$. The above lemma means that given an instance J of erasures in Z^n, the message S remains completely secret if and only if the submatrix $\mathbf{G}_{[n]\setminus J}$ corresponding to the revealed bits has full rank. Whenever this happens, the bits corresponding to the revealed positions in Z^n cycle through all $2^{n-|J|}$ possibilities in the linear code $\mathcal{C}$. That is, the projection of $\mathcal{C}$ onto the bit positions in $[n]\setminus J$ is the entire space $\mathbb{F}^{n-|J|}$. This immediately means that in each coset C_i of the code $\mathcal{C}$, the bits corresponding to the revealed bit-positions cycle through all possibilities. Therefore, given the observation Z^n, all cosets of $\mathcal{C}$ will have the same number of vectors that match with Z^n, which means that $\mathrm{I}(S; Z^n) = 0$.

Example 3.2. Consider the coset coding scheme using the linear code $\mathcal{C}$ defined by the following generator matrix $\mathbf{G}$

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

The code $\mathcal{C}$ has block length $n = 5$, dimension two and eight ($= 2^{5-2}$) cosets. Therefore, the coset coding scheme will take eight possible messages, each message $\mathbf{s}_i$ being

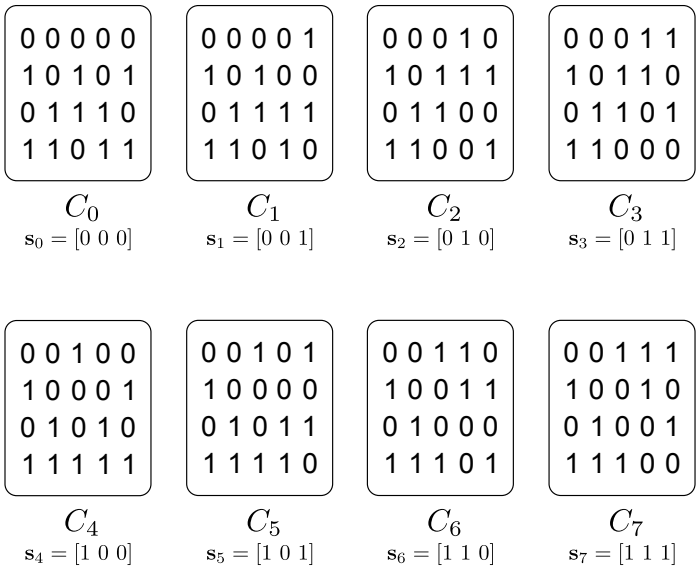


Figure 10: Example 3.2: An illustration of the cosets and the corresponding secret messages.

a 3-bit vector, and it will output a 5-bit vector X^n. The cosets of $\mathcal{C}$ and the corresponding secret messages are illustrated in Fig. 10.

Suppose the secret message $S = \mathbf{s}_3 = [0\ 1\ 1]$. To transmit this secret message, the coset coding scheme based on $\mathcal{C}$ will select the coset C_3 and output a random vector from it as X^n. In this example, suppose $X^n = [0\ 1\ 1\ 0\ 1]$. To illustrate the reasoning behind Lemma 3.1, consider two different cases.

- Suppose Z^n is the erased version of the output vector X^n with the first, second and fifth bits erased. That is, $J = \{1, 2, 5\}$ and $Z^n = [?\ ?\ 1\ 0\ ?]$. The submatrix corresponding to the revealed bits is

$$\mathbf{G}_{\{3,4\}} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

  and it has full rank. We can also see (Fig. 11) that for this case, each coset contains exactly one vector that matches with the observation $Z^n = [?\ ?\ 1\ 0\ ?]$. In this case, Z^n does not reveal which three-bit pattern is intended as the secret S.

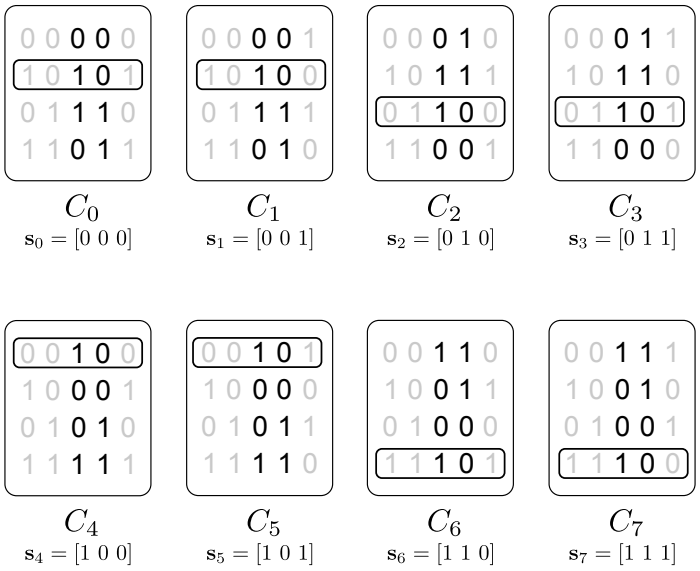


Figure 11: Example 3.2: The vectors matching the observation $Z^n = [?\ ?\ 1\ 0\ ?]$.

- Now suppose $Z^n = [0\ ?\ ?\ ?\ 1]$, that is, the second, third and fourth bits are erased. In this case, the submatrix of G corresponding to the revealed bits is

$$\mathbf{G}_{\{1,5\}} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$$

  and it does not have full rank. Moreover, only four out of the eight cosets match the observation Z^n. Therefore, some amount of information about S is revealed by Z^n. In particular, note from Fig. 12 that the last bit of S is leaked by Z^n. ▼

3.3 From Secrecy Codes to Channel Codes

3.3.1 Equivocation Analysis for the BEWC

Suppose the coset coding scheme using the linear code $\mathcal{C}$ is used to send a secret message over BEWC(ξ). Let us analyze the amount of information leaked to Eve through her observation Z^n. Let J be the index set of the erased bits in Eve's received vector. The vector J is a random vector dictated by the erasure probability

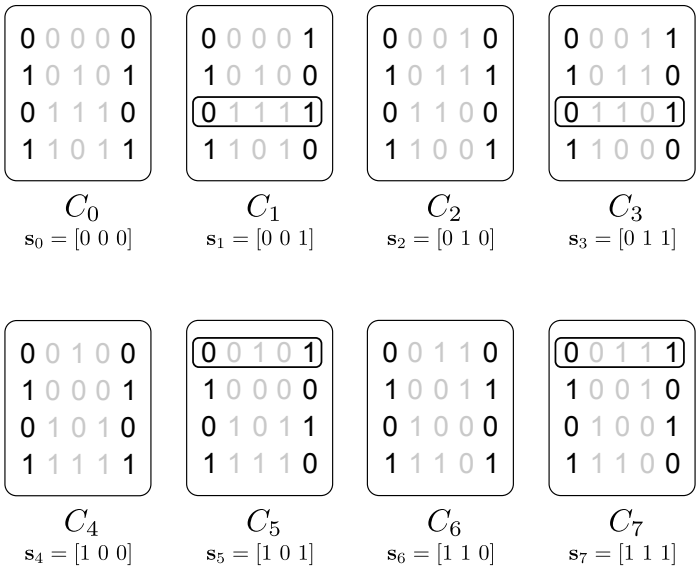


Figure 12: Example 3.2: The vectors matching the observation $Z^n = [0\ ?\ ?\ ?\ 1]$.

ξ. Let F be the event that $\mathbf{G}_{[n]\setminus J}$ has full-rank and $\bar{F}$ be the complementary event. We can now write the following.

$$\mathrm{I}(S; Z^n) = \Pr(F)\ \mathrm{I}(S; Z^n|F) \quad + \quad \Pr(\bar{F})\ \mathrm{I}(S; Z^n|\bar{F})$$

By Lemma 3.1, the conditional mutual information $\mathrm{I}(S; Z^n|F)$ is zero. The conditional mutual information $\mathrm{I}(S; Z^n|\bar{F})$ can be upper bounded by the entropy of the secret, that is

$$\mathrm{I}(S; Z^n|\bar{F}) \leq \mathrm{H}(S) = n\mathtt{R}$$

Therefore, we have

$$\mathrm{I}(S; Z^n) \leq n\ \mathtt{R}\ \Pr(\bar{F}) \tag{2}$$

3.3.2 A Channel Coding Problem

Now, consider the channel coding problem involving the binary erasure channel with erasure probability $1 - \xi$ and the dual of the code $\mathcal{C}$. The dual code $\mathcal{C}^\perp$ is defined as

the set of all n-bit vectors orthogonal to $\mathcal{C}$. That is,

$$\mathcal{C}^{\perp} \triangleq \{\mathbf{x} \in \mathbb{F}_2^n \ : \ \mathbf{x} \cdot \mathbf{c} = 0, \forall \mathbf{c} \in \mathcal{C}\}$$

where $\mathbf{x} \cdot \mathbf{c}$ is the dot product of the vectors $\mathbf{x}$ and $\mathbf{c}$. The code $\mathcal{C}^{\perp}$ is a $(n, n\mathtt{R})$ binary linear code.

Suppose a uniformly distributed $n\mathtt{R}$-bit message is encoded using $\mathcal{C}^{\perp}$ and the resulting n-bit codeword is transmitted over BEC$(1-\xi)$. Let $\hat{Z}^n$ be the output of the channel with $\hat{J}$ being the index set of the revealed bits (unerased bits) in the channel output. In the previous section, we transmitted n-bits through BEC(ξ) and denoted the index set of the erased bits by J. We can easily see that J and $\hat{J}$ are identically distributed random variables.

It can be immediately seen that the generator matrix $\mathbf{G}$ of the code $\mathcal{C}$ is also the parity-check matrix of the dual code $\mathcal{C}^{\perp}$. The parity-check equation for the encoded word $\hat{X}$ is given by

$$\begin{aligned} \mathbf{G} \ \hat{X}^T &= 0 \\ \Rightarrow \mathbf{G}_{\hat{J}} \ \hat{X}_{\hat{J}} \ + \ \mathbf{G}_{[n]\setminus\hat{J}} \ \hat{X}_{[n]\setminus\hat{J}} &= 0 \\ \Rightarrow \mathbf{G}_{[n]\setminus\hat{J}} \ \hat{X}_{[n]\setminus\hat{J}} &= \mathbf{G}_{\hat{J}} \ \hat{X}_{\hat{J}} \end{aligned} \tag{3}$$

Since $\hat{J}$ corresponds to the revealed bit-positions in $\hat{Z}^n$, the right hand-side of (3) is a known vector. The maximum-a-posteriori decoder for the erasure channel must solve (3) for the unknown vector $\hat{X}_{[n]\setminus\hat{J}}$. There is a unique solution to this equation if and only if the sub-matrix $\mathbf{G}_{[n]\setminus\hat{J}}$ has full rank. Let $\hat{F}$ be the even that $\mathbf{G}_{[n]\setminus\hat{J}}$ has full rank. Let us denote the probability of decoder failure under maximum a posteriori (MAP) decoding by $P_B^{\mathrm{MAP}}(\mathcal{C}^{\perp}, 1-\xi)$, where the first parameter denotes the code used over the BEC and the second parameter denotes the erasure probability of the BEC. We immediately note the following

$$\Pr(\hat{F}) = 1 - P_B^{\mathrm{MAP}}(\mathcal{C}^{\perp}, 1-\xi)$$

Since the event F in the previous section and the event $\hat{F}$ are identically distributed, we may substitute the above equation in (2) to get

$$\begin{aligned} \mathrm{I}(S; Z^n) &\leq n \ \mathtt{R} \ \Pr(\bar{F}) = n \ \mathtt{R} \ \Pr(\bar{\hat{F}}) \\ \Rightarrow \mathrm{I}(S; Z^n) &\leq n \ \mathtt{R} \ P_B^{\mathrm{MAP}}\left(\mathcal{C}^{\perp}, 1-\xi\right) \end{aligned} \tag{4}$$

For $k \in \mathbb{N}$, suppose we have a sequence of binary linear codes $(\mathcal{C}_{n_k})$ with block length n_k increasing monotonically as k increases, such that the codes have a constant rate $1 - \mathtt{R}$. In order to make the notation easier, we drop the subscript k and denote the sequence of codes by $\mathcal{C}_n$. Suppose we use these codes in the coset coding scheme to transmit a secret message over BEWC(ξ), for arbitrary $\xi \in [0, 1]$. We have the following sufficient condition for strong secrecy.

Lemma 3.3. *The coset coding scheme using the sequence $(\mathcal{C}_n)$ achieves strong secrecy on BEWC(ξ) whenever the asymptotic block-error probability of the dual sequence $(\mathcal{C}_n^{\perp})$ over BEC($1-\xi$) decays as*

$$P_B^{\mathrm{MAP}}\left(\mathcal{C}_n^{\perp}, 1-\xi\right) = \mathcal{O}\left(\frac{1}{n^2}\right)$$

From the union bound, we know that for any code over any channel, we have

$$\Pr(\text{bit-error}) \leq \Pr(\text{block-error}) \leq n \ \Pr(\text{bit-error})$$

We have the following sufficient condition using the above bound.

Lemma 3.4. *The coset coding scheme using the sequence $(\mathcal{C}_n)$ achieves strong secrecy on BEWC(ξ) whenever the asymptotic bit-error rate (BER) of the dual sequence $(\mathcal{C}_n^{\perp})$ over BEC($1-\xi$) decays as*

$$P_b^{\mathrm{MAP}}\left(\mathcal{C}_n^{\perp}, 1-\xi\right) = \mathcal{O}\left(\frac{1}{n^3}\right)$$

Note that the above sufficient condition involves the MAP decoder. Since MAP decoding is optimal, P_B^{MAP} can be upper bounded by the block-error probability under

any suboptimal decoder. Therefore, the duals of codes whose block-error probability under suboptimal decoding decays as $\mathcal{O}(1/n^2)$ will achieve strong secrecy on the BEWC when used in a coset coding scheme. In the next two chapters, we will describe two codes which have this property.

CHAPTER IV

SHORT-CYCLE-FREE LDPC CODES FOR STRONG SECRECY

The objective of our research is to design coding schemes to achieve strong secrecy on the BEWC model. In Chapter 3, we showed that this problem can be reduced to the problem of finding a sequence of binary linear block codes whose block-error probability on the BEC decays as $1/n^2$ as the block length n increases. In this chapter, we will discuss our strong secrecy result using short-cycle-free LDPC codes. We study the block-error probability of LDPC codes under belief propagation (BP) decoding by analyzing its *stopping sets*. The key result in this chapter is that LDPC codes with minimum variable node degree at least three and girth at least six (hence called "short-cycle-free") have a BP block-error probability that decays as $1/n^2$. The work discussed in this chapter was presented in [36, 37].

This chapter is organized as follows. In the first section, we give a quick overview of LDPC codes and discuss some of the results involving stopping sets of LDPC codes that form the basis of the our work. In the next section, we provide an overview of the key ideas in our result. In the third and fourth sections, we prove our main result. In the final section, we describe the strong secrecy region achieved by using short-cycle-free LDPC codes.

4.1 Fundamentals of LDPC Codes

LDPC codes are linear error-correcting codes introduced by Gallager in his thesis [38]. After their original discovery in the 1960's, these codes were forgotten for a few decades until their rediscovery in the 1990's. In this section, we will give a brief

$$\mathbf{H} = \begin{array}{c} \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 \end{matrix} \\ \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{array} \begin{matrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{matrix}$$

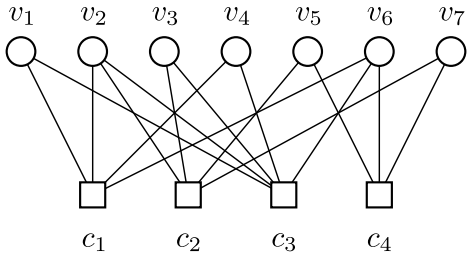


Figure 13: A parity-check matrix and its corresponding Tanner graph representation.

introduction to LDPC codes, with sufficient details to follow the remainder of this chapter. Most of the material in this section is based the notation and presentation from the book *Modern Coding Theory* by Richardson and Urbanke [34]. The material on stopping sets (§ 4.1.4) is based on the paper by Orlitsky, et al. [39].

Definition 4.1 (Tanner graph)**.** Given the $(n-k) \times n$ parity-check matrix $\mathbf{H}$ of a binary linear code $\mathcal{C}$, the *Tanner graph* corresponding to $\mathcal{C}$ is a bipartite multigraph G with the vertex bipartition $V(G) = V_v \cup V_c$ such that

- The set V_v has n *variable nodes*, with each node v_i representing a codeword bit x_i in $\mathcal{C}$.
- The set V_c has $(n-k)$ *check nodes*, with each node c_j representing a row $\mathbf{h}_j$ of the parity-check matrix $\mathbf{H}$ of $\mathcal{C}$.
- There is an edge between a variable node v_i and a check node c_j if and only if the codeword bit x_i corresponding to v_i appears in the parity-check equation corresponding to the row $\mathbf{h}_j$ represented by c_j. ▼

Fig. 13 shows a parity-check matrix and its corresponding Tanner graph. In this dissertation, we will allow Tanner graphs with multiple edges with the following equivalence—a multiple edge with odd multiplicity is equivalent to having a single edge between the two involved nodes and a multiple edge with even multiplicity is equivalent to having no edge between the two nodes. Following this convention, we will use the term *graph* to mean a multigraph with no loops.

4.1.1 LDPC Codes

Given a binary linear block code $\mathcal{C}$, the parity-check matrix $\mathbf{H}$ of $\mathcal{C}$ is not necessarily unique. This means that $\mathcal{C}$ has several Tanner graph representations. LDPC codes are a special class of linear codes with the property that for a given family of codes, there exists a Tanner graph family corresponding to these codes such that the number of edges in the Tanner graph family increases *linearly* with block length. The term *low-density* is applied to these codes because the number of edges in the Tanner graph sequence grows slowly (linearly) with the block length n; in other words, the *density* of ones in the parity-check matrix remains constant as n increases. We define an LDPC code family by defining its Tanner graph family, or specifically, the number of variable and check nodes in the Tanner graphs and their degrees.

Definition 4.2 (LDPC Code)**.** Given two polynomials

$$\lambda(x) = \sum_{i=\mathtt{l}_{\min}}^{\mathtt{l}_{\max}} \lambda_i x^{i-1}, \qquad \rho(x) = \sum_{j=\mathtt{r}_{\min}}^{\mathtt{r}_{\max}} \rho_j x^{j-1}$$

with $\lambda_i, \rho_j \in [0,1] \cap \mathbb{Q}$ and $\sum_i \lambda_i = \sum_j \rho_j = 1$, a (λ, ρ) *LDPC code of block length* n is defined as the binary linear block code which corresponds to a Tanner graph with n variable nodes such that the fraction of edges connected to degree-i variable nodes is λ_i, and the fraction of edges connected to degree-j check nodes is ρ_j. The polynomial pair (λ, ρ) is called the *degree distribution pair (DDP)* of the LDPC code. ▼

Definition 4.3 (Regular and Irregular LDPC Codes)**.** Given two positive integers

$\mathtt{l}$ and $\mathtt{r}$ with $\mathtt{r} \geq \mathtt{l}$, an $(\mathtt{l}, \mathtt{r})$-*regular* LDPC code is one in which all variable nodes have the same degree $\mathtt{l}$ and all check nodes have the same degree $\mathtt{r}$, i.e., $\lambda(x) = x^{\mathtt{l}-1}$, $\rho(x) = x^{\mathtt{r}-1}$. An LDPC code that is not regular is called an *irregular* LDPC code. The Tanner corresponding to a regular (irregular) LDPC code is called a regular (irregular) Tanner graph. ▼

While defining Tanner graphs in terms of the fraction with degree-i variable nodes and the fraction with degree-j check nodes may appear straightforward, the above indirect description of the degrees is employed because the polynomials $\lambda(x)$ and $\rho(x)$ play an important role in the performance analysis of LDPC codes. The description of the degrees using the polynomials (λ, ρ) is often called the *edge-perspective* degree distribution. If we denote the edge set of the Tanner graph by E, then the number of variable nodes with degree i is $n_i = |E|\lambda_i/i$ and the number of check nodes with degree j is $m_j = |E|\rho_j/j$. The number of variable nodes n and the number of check nodes m in the Tanner graph are given by

$$n = |E| \sum_i \frac{\lambda_i}{i} = |E| \int_0^1 \lambda(x) \mathrm{d}x$$
$$m = |E| \sum_j \frac{\rho_j}{j} = |E| \int_0^1 \rho(x) \mathrm{d}x$$

The fraction L_i of variable nodes with degree i and the fraction R_j of check nodes with degree j are given by

$$L_i = \frac{\lambda_i}{i} \frac{1}{\int_0^1 \lambda(x) \mathrm{d}x}, \qquad R_j = \frac{\rho_j}{j} \frac{1}{\int_0^1 \rho(x) \mathrm{d}x}$$

The *node-perspective* degree distribution polynomials are defined as

$$L(x) = \sum_{i=\mathtt{l}_{\min}}^{\mathtt{l}_{\max}} L_i x^i$$
$$R(x) = \sum_{j=\mathtt{r}_{\min}}^{\mathtt{r}_{\max}} R_j x^j$$

Note that the definition of $L(x)$ is not quite analogous to the definition of $\lambda(x)$ because in the former we multiply L_i by x^i, whereas in the latter we multiply λ_i by x^{i-1}. The definitions of $R(x)$ and $\rho(x)$ differ in the same point.

Clearly, an LDPC code with block length n and node-perspective degree distribution pair (L, R) must be such that nL_i is an integer for all i. Therefore, the code is defined only when n is an integral multiple of some integer a such that $aL_i \in \mathbb{N}$ for all i.

The *design rate* $\mathtt{R}^*$ of the LDPC code is defined as

$$\mathtt{R}^* = 1 - \frac{m}{n} = 1 - \frac{\int_0^1 \rho(x)\mathrm{d}x}{\int_0^1 \lambda(x)\mathrm{d}x}$$

Clearly, $\lambda(x)$ and $\rho(x)$ should be such that $\mathtt{R}^* \in [0, 1]$ for this discussion to be meaningful. The actual rate $\mathtt{R}$ of the code can be higher than $\mathtt{R}^*$, and we have $\mathtt{R} = \mathtt{R}^*$ if and only if the m parity-check equations represented in the Tanner graph are linearly independent.

4.1.2 The Standard Ensemble of LDPC Codes

When classical error-correcting codes are studied in coding theory, a single code is considered and its performance over a noisy channel is studied. In contrast to this, a single instance of an LDPC code is not studied; instead, an ensemble of LDPC codes with a given degree distribution and block length is considered, and the average performance of the codes in this ensemble is studied. This is done because studying the properties of a single LDPC code is very hard. Moreover, certain LDPC code ensembles are such that most of the codes in it closely follow the average performance. In such a case, we say that there is *concentration* around the average performance. While studying LDPC codes, we usually consider the *standard ensemble*, which is defined subsequently.

Given a degree distribution pair (DDP) (λ, ρ) and block length n, let m be the number of check nodes in a (λ, ρ) Tanner graph with n variable nodes. A fraction L_i of the n variable nodes have degree i and a fraction R_j of the m check nodes will have degree j. To construct the standard ensemble of Tanner graphs, denoted by $\mathcal{G}(n, \lambda, \rho)$, we create n variable nodes and m check nodes such that nL_i of the variable

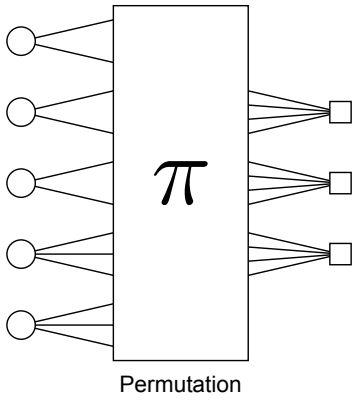


Figure 14: The construction of the standard ensemble $\mathcal{G}(5, \frac{1}{2}x + \frac{1}{2}x^2, x^3)$.

nodes have i "sockets" each and mR_j of the check nodes have j sockets each (for all possible i and j). We label the variable node sockets as $1, 2, \ldots, |E|$ and the check node sockets as $1, 2, \ldots, |E|$. This labeling is done in some arbitrary manner, but remains fixed throughout the construction of $\mathcal{G}(n, \lambda, \rho)$. Consider a permutation function $\pi : \{1, 2, \ldots, |E|\} \to \{1, 2, \ldots, |E|\}$. The Tanner graph in $\mathcal{G}(n, \lambda, \rho)$ corresponding to the permutation π is constructed by connecting the i^{th} variable-node socket to the $\pi(i)^{\text{th}}$ check-node socket with an edge, for $i \in \{1, 2, \ldots, |E|\}$. The ensemble $\mathcal{G}(n, \lambda, \rho)$ is constructed by considering all possible permutation functions π. Therefore, this ensemble will have $|E|!$ Tanner graphs. For example, Fig. 14 depicts the construction of the ensemble $\mathcal{G}(5, \frac{1}{2}x + \frac{1}{2}x^2, x^3)$, where the permutation function π is varied over all 12! different possibilities.

Note that $\mathcal{G}(n, \lambda, \rho)$ will contain Tanner graphs with parallel (multiple) edges. In practice, we will never use such LDPC codes. However, we allow the possibility of multiple edges because the ensemble becomes much easier to study. Typically, we will study the probability that a randomly chosen graph from $\mathcal{G}(n, \lambda, \rho)$ has a particular local structure. Given a subset of socket connection pairs (local structure), the number of graphs that have this structure is precisely equal to the number of ways we can connect the remaining sockets. The combinatorial analysis of the number of

ways to connect the remaining sockets is much easier if we allow the possibility of multiple edges.

4.1.3 Belief Propagation Decoding of LDPC Codes Over the BEC

One of the main advantages of LDPC codes is that they can be decoded efficiently using a class of iterative algorithms called *message-passing* algorithms. A message-passing algorithm can be visualized in terms of a sparse network connecting simple distributed computing hardware [40, Ch. 16]. Messages are passed between vertices in the Tanner graph along the edges with the property that the outgoing message sent by a vertex along a particular edge depends on the incoming message on all other edges adjacent to that vertex and the channel observation (if any). The BP algorithm is one such message-passing algorithm where the messages are probabilities or *beliefs* of what codeword was sent on the channel.

Consider an LDPC code $\mathcal{C}$ with the Tanner graph representation G transmitted over a BEC. That is, a randomly selected codeword from $\mathcal{C}$ is transmitted and the received word is a vector in $\{0, 1, ?\}$, where ? represents an erased bit. The BP decoding algorithm for recovering the erased bits is equivalent to the following algorithm.

- The messages sent along the edges are from the alphabet $\{0, 1, ?\}$.
- **Variable node processing**: An erasure message '?' is sent along an edge if the channel observation for the variable node is an erasure and all other incoming messages are also erasures. Otherwise, the channel observation or an incoming non-erasure message from any of the other edges is sent. In the latter case, all incoming non-erasure messages at a variable node will be identical; if the channel observation is not an erasure, they will be identical to the channel observation.
- **Check node processing**: An erasure message is sent along an edge if any of

the other incoming messages are erasures. Otherwise, the sum (over $\mathbb{F}_2$) of all other incoming messages is sent.

For general channels, we must specify a *schedule* for the BP algorithm, i.e., we need to specify the sequence in which the node processing is done for all the nodes in the Tanner graph. In the case of the BP algorithm for the BEC, any reasonable schedule will yield the same performance. For the sake of clarity, we assume that we use the *flooding schedule*.

The flooding schedule is the most commonly used schedule for BP decoding and it operates as follows. In the zeroth iteration, the variable nodes send their received value along all their edges *simultaneously*. All further iterations consist of two steps. In the first step, all check nodes are processed simultaneously. In the second step, all variable nodes are processed simultaneously. The flooding schedule is so called because the messages are released on to all the edges at the same time (i.e., "flooded"). In practical applications, only a finite number of iterations of the BP algorithm are performed. The BP algorithm for erasures is a special case where the number of "errors" (strictly speaking, undecoded bits) decreases monotonically as the number of iterations increases; the BP decoder will reach a certain stage where no more decoding is possible. In this chapter, we will only consider the operation of the BP decoder until this saturation point.

The erasure BP decoding algorithm with the flooding schedule is equivalent to the following iterative graphical evolution algorithm.

1. Let G^* be the subgraph of the Tanner graph G induced by the variable nodes associated with the unknown codeword bits, and their neighboring check nodes. If G^* is an empty graph, then declare decoding success and exit.

2. Find all variable nodes in G^* that are connected to degree-one check nodes (in G^*). If there are no such nodes, then declare decoding failure and exit.

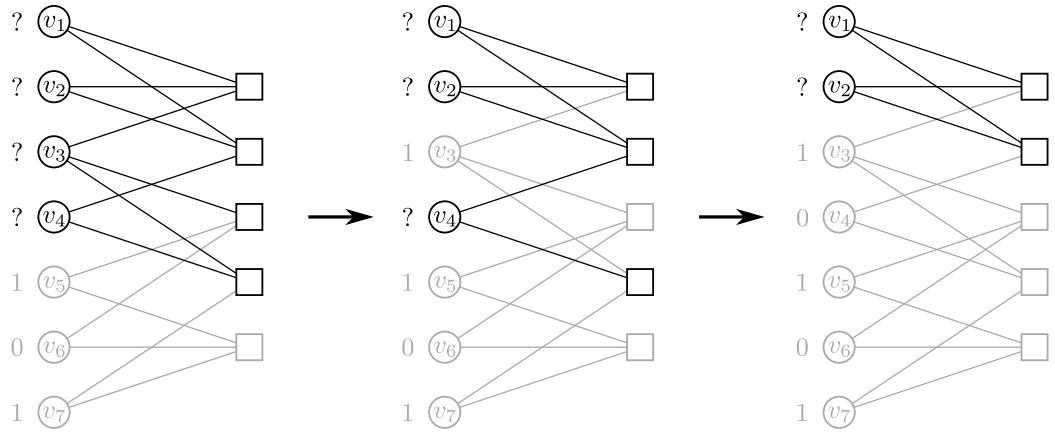


Figure 15: An example of the graphical evolution under BP decoding. The edges and vertices in black denote the subgraph G^* induced by the unsolved variable nodes and their neighbors.

3. Solve the codeword bits associated with the variable nodes found in the previous step.

4. Go to Step 1.

From the above description of BP decoding, we can see that whenever BP decoding failure occurs, the unknown codeword bits at the last stage induce a subgraph G^* such that all the check nodes in G^* have degree two or more.

4.1.4 Stopping Sets

Definition 4.4 (Stopping Set)**.** Let G be a Tanner graph with variable-node set V_v and check-node set V_c. We call a set $S \in V_v$ a *stopping set* if the subgraph G^* induced by S and its neighbors (in V_c) has no degree-one check nodes. ▼

For example, in Fig. 15, the set $\{v_1, v_2\}$ is a stopping set. For a given LDPC code $\mathcal{C}$, we denote the collection of all stopping sets in its Tanner graph G by $\mathbb{S}(\mathcal{C})$. The BP decoder for LDPC codes with erasures fails if and only if the erasure pattern contains a stopping set. If we denote the probability of block error of the code $\mathcal{C}$ transmitted

over BEC(ε) under BP decoding by $P_B^{\mathrm{BP}}(\mathcal{C}, \varepsilon)$, we have

$$P_B^{\mathrm{BP}}(\mathcal{C}, \varepsilon) = \Pr\left(\exists S \subseteq V_e : S \in \mathbb{S}(\mathcal{C})\right)$$

where the random variable V_e denotes the set of all variable nodes in G that correspond to the bits erased during channel transmission. Note that in the above, we define the block-error probability for a fixed code $\mathcal{C}$. In this dissertation, we usually deal with the case where $\mathcal{C}$ itself is random. If the graph G is chosen uniformly at random from the standard ensemble $\mathcal{G}(n, \lambda, \rho)$, then the average probability of block error can be calculated as

$$\mathbb{E}\left(P_B^{\mathrm{BP}}(\mathcal{C}, \varepsilon)\right) = \mathbb{E}\left(\Pr\left(\exists S \subseteq V_e : S \in \mathbb{S}(\mathcal{C})\right)\right)$$

where the expectation $\mathbb{E}$ is taken over all $G \in \mathcal{G}(n, \lambda, \rho)$. From the above equation, we see that by characterizing the asymptotic distribution of stopping sets, we can completely characterize the average probability of block error of the standard ensemble of LDPC codes. We will now discuss some of the definitions and results from [39].

For an arbitrary integer $s \in \{0, 1, \ldots, n\}$, the *average stopping set distribution* of the standard ensemble is defined as

$$\mathcal{E}(s) = \mathbb{E}\left(|\{S \in \mathbb{S}(\mathcal{C}) : |S| = s\}|\right)$$

In other words, $\mathcal{E}(s)$ is the average number of stopping sets of size s in the standard ensemble.

For any rational $\alpha \in [0, 1]$, it is assumed that there exists a sequence $(n_k)_{k \in \mathbb{N}}$ of increasing block lengths such that $\mathcal{E}(\alpha n_k) > 0$ for all n_k. Under this assumption, the *normalized stopping set distribution* can be defined as

$$\gamma(\alpha) = \lim_{k \to \infty} \frac{1}{n_k} \log_2 \mathcal{E}(\alpha n_k)$$

It can be shown that $\gamma(\alpha)$ is continuous over the set of rationals in $[0, 1]$ and hence, it can be extended to a continuous function over $[0, 1]$.

One of the main results in [39] is that below a certain "error-floor threshold", the average block-error probability of the standard ensemble of LDPC codes decays polynomially over the BEC.

Theorem 4.5 ([39, Thm. 16]). *For a code $\mathcal{C}$ chosen uniformly at random from the ensemble $\mathcal{G}(n, \lambda, \rho)$ with $\mathtt{l}_{\min} \geq 2$, if $\varepsilon < \varepsilon_{\text{ef}}$, then*

$$\mathbb{E}\left(P_B^{\text{BP}}(\mathcal{C}, \varepsilon)\right) = \Theta\left(\frac{\varepsilon}{n^{\lceil \frac{\mathtt{l}_{\min}}{2} \rceil - 1}}\right)$$

as $\varepsilon \to 0$ and $n \to \infty$.

The parameter ε_{ef} for a given DDP (λ, ρ) is defined [39, §V] as

$$\varepsilon_{\text{ef}} \triangleq \sup\left\{\varepsilon : \max_{\alpha \in [0,\varepsilon]} \left(\gamma(\alpha) + (1-\alpha) h\left(\frac{\varepsilon - \alpha}{1-\alpha}\right) - h(\varepsilon)\right) \leq 0\right\}$$

where $h(x) \triangleq -x \log_2(x) - (1-x)\log_2(1-x)$ is the binary entropy function.

4.2 An Overview of Results and Intuition

The research presented in this chapter uses the results of Orlitsky, et al. [39] as a starting point. One of the key ideas in [39] is the use of the union bound to upper bound the average block-error probability of an ensemble of LDPC codes. Given a countable set of events $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \ldots\}$, the union bound says that the probability of at least one of the events happening is less than the sum of the individual probabilities. That is,

$$\Pr(\mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3 \cup \cdots) \leq \sum_i \Pr(\mathcal{A}_i)$$

The average block-error probability of an ensemble of LDPC codes is given by the probability that the erasure pattern V_e contains a stopping set $S \in \mathbb{S}(\mathcal{C})$.

$$P_B^{\text{BP}}(\mathcal{C}, \varepsilon) = \Pr(\exists S \subseteq V_e : S \in \mathbb{S}(\mathcal{C}))$$

The event that V_e contains a stopping set is the union of the events $\mathcal{A}_i$ defined as

$$\begin{aligned} \mathcal{A}_i &\triangleq V_e \text{ contains a size-}i \text{ stopping set} \\ \Rightarrow \Pr(\mathcal{A}_i) &= \Pr\{\exists S \subseteq V_e : S \in \mathbb{S}(\mathcal{C}), |S| = i\} \\ &= |\{S \in \mathbb{S}(\mathcal{C}) : |S| = i\}|\, \varepsilon^i \end{aligned}$$

Therefore, it can be seen that

$$\begin{aligned} \mathbb{E}\left(P_B^{\mathrm{BP}}(\mathcal{C}, \varepsilon)\right) &\leq \mathbb{E}\left(\sum_i \Pr(\mathcal{A}_i)\right) \\ &= \sum_i \varepsilon^i \; \mathbb{E}\left(|\{S \in \mathbb{S}(\mathcal{C}) : |S| = i\}|\right) \\ &= \sum_{i=1}^{n} \varepsilon^i \, \mathcal{E}(i) \end{aligned} \tag{5}$$

Orlitsky, et al. [39] showed that whenever $\varepsilon < \varepsilon_{\mathrm{ef}}$, the above summation decays to zero as n increases. Moreover, the asymptotically significant term in (5) is the one corresponding to size-1 stopping sets, namely, $\varepsilon\,\mathcal{E}(1)$, and this term decays as

$$\mathcal{O}\left(\frac{1}{n^{\lceil \frac{1_{\min}}{2} \rceil - 1}}\right)$$

4.2.1 Expurging Graphs with Short Cycles

In (5), the summation is over the contribution of stopping sets of size one or more. The intuition behind our work is based on the following observation (§4.4.1). If we sum the terms $\varepsilon^i\,\mathcal{E}(i)$ for stopping sets of size k or greater, where k is a positive integer, then the asymptotically significant contribution is the one corresponding to $i = k$ and this term decays as

$$\mathcal{O}\left(\frac{1}{n^{\lceil \frac{1_{\min}}{2} k \rceil - k}}\right)$$

This means that $\sum_{i=k}^{n} \varepsilon^i\,\mathcal{E}(i)$ decays at the above rate, which is faster than that of the summation in (5). This gives rise to the intuition that by removing stopping sets of size less than k, we will have a faster decay of the block-error rate. In other words,

we must expurgate Tanner graphs with stopping sets of size less than k from the standard ensemble $\mathcal{G}(n, \lambda, \rho)$.

Instead of expurgating graphs with small stopping sets, we consider the expurgation of graphs with short cycles from the standard ensemble. We do this because the girth of graphs is a well studied property with a lot of mathematical results. Moreover, there is a fundamental relationship between girth and stopping sets.

Lemma 4.6. *For $\mathtt{l}_{\min} \geq 2$, any graph $G \in \mathcal{G}(n, \lambda, \rho)$ with girth $2k$ or more will not have any stopping sets of size less than k.*

Proof. We will prove this by contradiction. Assume that the graph G has a stopping set of size $s < k$. Consider the subgraph G^* induced by this stopping set and its check-node neighbors. All the check nodes in G^* have degree two or more. Since $\mathtt{l}_{\min} \geq 2$, all the variable nodes in G^* also have degree two or more. This means that G^* is not a tree and should therefore contain a cycle. Since there are only s variable nodes in G^* and G^* is bipartite, the length of the cycle cannot be more than $2s$, which, in turn, is less than $2k$. Since G^* is a subgraph of G, this is a contradiction. □

So far, we have established that removing graphs with short cycles from $\mathcal{G}(n, \lambda, \rho)$ will give us an expurgated ensemble that does not have any small stopping sets. We now consider the probability of block-error for the expurgated ensemble. Suppose a code $\mathcal{C}_1$ is chosen at random from the expurgated ensemble and transmitted over BEC(ε). The expected BP block error probability is given by

$$\begin{aligned}
\mathbb{E}\left(P_B^{\mathrm{BP}}(\mathcal{C}_1, \varepsilon)\right) &= \sum_i \varepsilon^i \ \mathbb{E}\left(|\{S \in \mathbb{S}(\mathcal{C}_1) : |S| = i\}|\right) \\
&= \sum_{i=1}^{n} \varepsilon^i \, \mathcal{E}_1(i) \\
&= \sum_{i=k}^{n} \varepsilon^i \, \mathcal{E}_1(i) \qquad (6)
\end{aligned}$$

where $\mathcal{E}_1(i) \triangleq \mathbb{E}\left(|\{S \in \mathbb{S}(\mathcal{C}_1) : |S| = i\}|\right)$. The final equality in (6) is because the expurgated ensemble has no stopping sets of size less than k. So far, we know that

$$\sum_{i=k}^{n} \varepsilon^i \, \mathcal{E}(i) = \mathcal{O}\left(\frac{1}{n^{\lceil \frac{1}{2}\min k \rceil - k}}\right) \tag{7}$$

This does not necessarily mean that the expression in (6) decays at the same rate. This is because we have not shown that $\mathcal{E}_1(i)$ and $\mathcal{E}(i)$ are equal. In fact, explicit computation of $\mathcal{E}_1(i)$ is difficult because the combinatorial analysis of the expurgated ensemble is hard. We get around this by showing that we can bound $\mathcal{E}_1(i)$ using $\mathcal{E}(i)$.

4.2.2 Bounding Expectations over the Expurgated Ensemble

We have two codes $\mathcal{C}$ and $\mathcal{C}_1$ selected at random. The code $\mathcal{C}$ is uniformly sampled from $\mathcal{G}(n, \lambda, \rho)$, the standard ensemble, and $\mathcal{C}_1$ is uniformly sampled from $\mathcal{G}_{2k-2}(n, \lambda, \rho)$, the ensemble consisting of Tanner graphs of girth at least $2k$. Let f be some function that maps a code to a non-negative real number. We have

$$\mathbb{E}\big(f(\mathcal{C}_1)\big) = \mathbb{E}\big(f(\mathcal{C}) \mid \mathcal{C} \in \mathcal{G}_{2k-2}(n, \lambda, \rho)\big)$$

Also, we observe that

$$\begin{aligned}
\mathbb{E}\big(f(\mathcal{C})\big) &= \mathbb{E}\big(f(\mathcal{C}) \mid \mathcal{C} \in \mathcal{G}_{2k-2}(n, \lambda, \rho)\big) \Pr\big(\mathcal{C} \in \mathcal{G}_{2k-2}(n, \lambda, \rho)\big) \\
&\quad + \mathbb{E}\big(f(\mathcal{C}) \mid \mathcal{C} \notin \mathcal{G}_{2k-2}(n, \lambda, \rho)\big) \Pr\big(\mathcal{C} \notin \mathcal{G}_{2k-2}(n, \lambda, \rho)\big) \\
&\overset{\text{a}}{\geq} \mathbb{E}\big(f(\mathcal{C}) \mid \mathcal{C} \in \mathcal{G}_{2k-2}(n, \lambda, \rho)\big) \Pr\big(\mathcal{C} \in \mathcal{G}_{2k-2}(n, \lambda, \rho)\big) \\
&= \mathbb{E}\big(f(\mathcal{C}_1)\big) \, \frac{|\mathcal{G}_{2k-2}(n, \lambda, \rho)|}{|\mathcal{G}(n, \lambda, \rho)|}
\end{aligned}$$

Here, (a) follows from the fact that $f(\mathcal{C})$ is always non-negative. We then show (Lemma 4.9) that for a given k, λ and ρ, we have

$$\frac{|\mathcal{G}_{2k-2}(n, \lambda, \rho)|}{|\mathcal{G}(n, \lambda, \rho)|} \geq p > 0$$

for large enough n and some positive number p. Therefore,

$$\mathbb{E}\left(f(\mathcal{C})\right) \geq p\,\mathbb{E}\left(f(\mathcal{C}_1)\right)$$
$$\Rightarrow \mathbb{E}\left(f(\mathcal{C}_1)\right) \leq \frac{1}{p}\,\mathbb{E}\left(f(\mathcal{C})\right)$$

We require $p > 0$ in order to write the last step in the above. By substituting $f(\mathcal{C}) = |\{S \in \mathbb{S}(\mathcal{C}) : |S| = i\}|$, we obtain

$$\mathcal{E}_1(i) \leq \frac{1}{p}\,\mathcal{E}(i)$$

4.3 The Asymptotic Fraction of Short-Cycle-Free LDPC Codes

When we study LDPC codes, we usually consider the standard ensemble of Tanner graphs. As discussed in §4.1.2, this ensemble contains Tanner graphs that have parallel edges, i.e., cycles of length two. Given a DDP (λ, ρ) and an even number $g > 0$, we show that the asymptotic fraction of Tanner graphs with girth greater than g in the standard ensemble $\mathcal{G}(n, \lambda, \rho)$ is positive, i.e.,

$$\frac{|\mathcal{G}_g(n, \lambda, \rho)|}{|\mathcal{G}(n, \lambda, \rho)|} > 0$$

for large enough n. Here, $\mathcal{G}_g(n, \lambda, \rho)$ denotes the subset of $\mathcal{G}(n, \lambda, \rho)$ that consists of graphs with girth greater than g. Strictly speaking, we show that this result is true for a given (λ, ρ) and $g > 0$ for a *particular* sequence of increasing block lengths. We conjecture that the result holds without any restrictions on the sequence of block lengths. However, the conjecture is not required for the results in this chapter.

4.3.1 Short-Cycle-Free Regular Bipartite Graphs

The result in this section hinges on the following result from the graph theory community, which states that for any even number $g > 0$, the asymptotic fraction of d-regular bipartite graphs with girth greater than g is positive.

Lemma 4.7 (McKay, et al. [41, Cor. 3])**.** *Let n, g be even positive integers and $d \geq 3$ be an integer. As n grows, let $(d-1)^{2g-1} = o(n)$. Then, the fraction of (labeled) d-regular bipartite graphs on n vertices with girth greater than g is*

$$\exp\left(-\sum_{s=1}^{g/2}\frac{(d-1)^{2s}}{2s} + o(1)\right)$$

as $n \to \infty$.

While the above lemma is true when d and g vary with n, we will consider only the special case where d and g are constant with n.

In the Lemma 4.7, McKay, et al. [41] consider the ensemble of vertex-labeled d-regular bipartite graphs in n vertices. Let us denote this ensemble by $\mathcal{G}^*(n,d)$. To build a graph in $\mathcal{G}^*(n,d)$, we first take n vertices, half of them on the *left* side and half of them on the *right* side, and assign them some arbitrary (but fixed) labels. In the second step, we add edges such that each vertex has exactly d edges, with each edge connecting a left vertex to a right vertex while allowing the possibility of parallel edges. By varying the second step, we generate all the graphs in $\mathcal{G}^*(n,d)$.

The ensemble $\mathcal{G}^*(n,d)$ is related to the standard ensemble of LDPC codes given by $\mathcal{G}(n/2, x^{d-1}, x^{d-1})$—they both consist of d-regular bipartite graphs in n vertices. In the ensemble $\mathcal{G}(n/2, x^{d-1}, x^{d-1})$, we label the *sockets* of the vertices instead of the vertices themselves. Given a socket labeling, we can create an ordering of the labels such that all sockets of a node form a contiguous cluster in the ordering. In this sense, the socket labeling in $\mathcal{G}(n/2, x^{d-1}, x^{d-1})$ naturally induces a vertex labeling (ordering). Let $f : \mathcal{G}(n/2, x^{d-1}, x^{d-1}) \to \mathcal{G}^*(n,d)$ be a function that takes a graph from the standard ensemble, and assigns the natural labels for the vertices and removes the socket labels to create a graph in the vertex-labeled ensemble. For example, Fig. 16 illustrates a socket-labeled graph G and its vertex-labeled counterpart $f(G)$.

Clearly, the map f is a surjection, i.e., $f\left(\mathcal{G}(n/2, x^{d-1}, x^{d-1})\right) = \mathcal{G}^*(n,d)$. We observe that since f only changes the labels and does not affect the structure of the

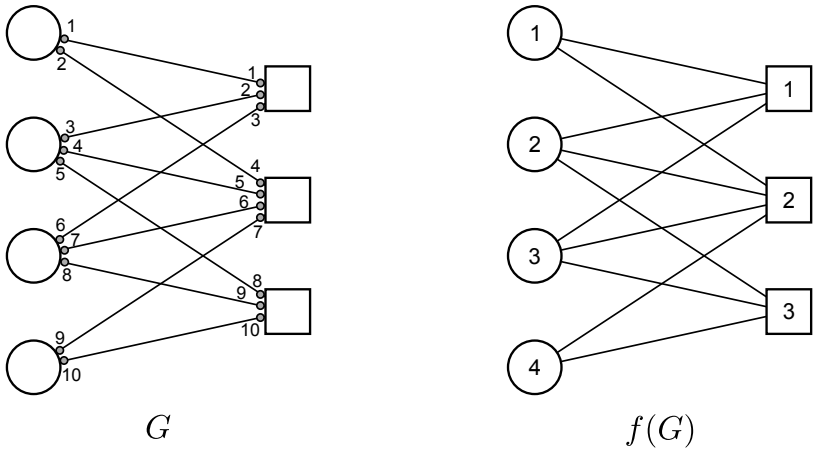


Figure 16: A socket-labeled graph G and its vertex labeled counterpart $f(G)$.

graph in any way; in particular,

$$\text{girth}(G) = \text{girth}\big(f(G)\big)$$

Therefore, we have

$$f\left(\mathcal{G}_g(n/2, x^{d-1}, x^{d-1})\right) = \mathcal{G}_g^*(n,d)$$

Here, $\mathcal{G}_g^*(n,d)$ denotes the subset of $\mathcal{G}^*(n,d)$ that consists of graphs with girth greater than g.

Consider the inverse image of any graph G in $\mathcal{G}^*(n,d)$. As long as G has no parallel edges, it has exactly $(d!)^n$ elements, i.e., $|f^{-1}(G)| = (d!)^n$. This is because any given vertex has d sockets, which can be labeled in $d!$ different ways. There are n vertices, so the number of graphs in $\mathcal{G}(n/2, x^{d-1}, x^{d-1})$ that map to G is exactly $(d!)^n$. However, the case where G has multiple edges is a little more complicated. It can be easily seen that $(d!)^n$ is an overcount of the number of graphs in $f^{-1}(G)$. If $M_1, M_2, \ldots$ is the multiplicity of the parallel edges in G, then we observe that

$$\left|f^{-1}(G)\right| = \frac{(d!)^n}{\prod_i M_i!}$$

In any case, we have $|f^{-1}(G)| \leq (d!)^n$. From the above discussion, we note the

following.

$$|\mathcal{G}^*(n,d)| \ (d!)^n \leq \left|\mathcal{G}(n/2, x^{d-1}, x^{d-1})\right| \tag{8}$$

$$\left|\mathcal{G}_g^*(n,d)\right| \ (d!)^n = \left|\mathcal{G}_g(n/2, x^{d-1}, x^{d-1})\right| \tag{9}$$

Note that in (9), we have an equality because the restriction 'girth $> g$' gets rid of graphs with parallel edges. We now have

$$\begin{aligned}\frac{|\mathcal{G}_g(n/2, x^{d-1}, x^{d-1})|}{|\mathcal{G}(n/2, x^{d-1}, x^{d-1})|} &\geq \frac{\left|\mathcal{G}_g^*(n,d)\right|}{|\mathcal{G}^*(n,d)|} \\ &\overset{\text{a}}{=} \exp\left(-\sum_{s=1}^{g/2} \frac{(d-1)^{2s}}{2s} + o(1)\right)\end{aligned}$$

where (a) follows from Lemma 4.7.

Corollary 4.8. *Let g, n be positive even numbers and let $d \geq 3$ be an integer. Let d and g remain constant as $n \to \infty$. Then, the fraction of graphs in the ensemble $\mathcal{G}(n/2, x^{d-1}, x^{d-1})$ with girth greater than g is at least*

$$\exp\left(-\sum_{s=1}^{g/2} \frac{(d-1)^{2s}}{2s} + o(1)\right)$$

as $n \to \infty$. In particular, the fraction is bounded away from zero for large n.

4.3.2 Short-Cycle-Free Irregular Tanner Graphs

In Corollary 4.8, we showed that short-cycle-free (d,d)-regular Tanner graphs form an asymptotically significant fraction of the graphs in the corresponding standard ensemble. In this section, we show a similar result for (λ, ρ) irregular Tanner graph ensembles—for a certain sequence $(n_k)_{k \in \mathbb{N}}$ of block lengths, we show that the asymptotic fraction of the graphs in $\mathcal{G}(n_k, \lambda, \rho)$ with girth greater than g is positive.

Lemma 4.9. *Let (λ, ρ) be a DDP and let $g > 0$ be an integer that remains constant. There exists an increasing sequence (n_k) of positive integers such that the fraction of*

graphs with girth greater than g in $\mathcal{G}(n_k, \lambda, \rho)$ is bounded away from zero as $k \to \infty$. That is,

$$\frac{|\mathcal{G}_g(n_k, \lambda, \rho)|}{|\mathcal{G}(n_k, \lambda, \rho)|} > 0 \qquad as\ k \to \infty$$

Proof. Let d be the least common multiple of all the vertex degrees in the graph, i.e.,

$$d = \mathrm{LCM}\left(\{i : \lambda_i > 0\} \cup \{j : \rho_j > 0\}\right)$$

If the above calculation gives $d = 2$, set $d = 4$. Clearly, $d > 2$ and it is a function of only λ and ρ. Let a be the smallest positive integer such that

$$\frac{aL_i}{d}, \frac{aR_j}{d} \in \mathbb{N}, \quad \forall i, j$$

Here, L_i is the fraction of variable nodes with degree i and R_j is the fraction of check nodes with degree j. Consider the sequence of block lengths $(n_k)_{k \in \mathbb{N}}$ given by $n_k = ak$. By choosing a in a specific manner, we have made sure that the number of degree-i variable nodes and the number of degree-j check nodes in $\mathcal{G}(n_k, \lambda, \rho)$ are both divisible by d for all i and j.

We now define a node-grouping map, which maps the graphs in $\mathcal{G}(n_k, \lambda, \rho)$ to (d, d)-regular Tanner graphs. Given a graph G from the former ensemble, we group d/i of the degree i variable nodes to get one variable node of degree d. If we do this for all the variable node degrees, we will have a left-regular Tanner graph with left degree d. Similarly, we can repeat this process for the check nodes to get a d-regular Tanner graph on n_k^* variable nodes and n_k^* check nodes, where

$$n_k^* = \sum_{i=2}^{\mathtt{l}_{\max}} \frac{i}{d} n_k L_i = \sum_{j=2}^{\mathtt{r}_{\max}} \frac{j}{d} n_k R_i$$

Fig. 17 shows an example of the node grouping map from $\mathcal{G}(4, x, x^3)$ to $\mathcal{G}(2, x^3, x^3)$. In this node grouping process, we preserve the number of edges (sockets) by allowing the possibility of multiple edges. We also preserve the socket labels while grouping

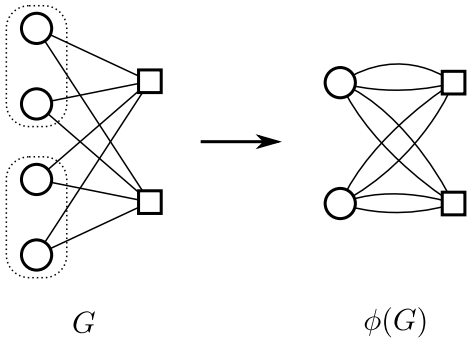


Figure 17: An example of the node grouping process. For simplicity, the socket labels are omitted.

the nodes. This directly means that the number of graphs in the ensembles $\mathcal{G}(n_k, \lambda, \rho)$ and $\mathcal{G}(n_k^*, x^{d-1}, x^{d-1})$ are equal, i.e.,

$$|\mathcal{G}(n_k, \lambda, \rho)| = \left|\mathcal{G}(n_k^*, x^{d-1}, x^{d-1})\right| \tag{10}$$

Let us denote the node grouping map by $\phi : \mathcal{G}(n_k, \lambda, \rho) \to \mathcal{G}(n_k^*, x^{d-1}, x^{d-1})$. By the socket preserving property of ϕ, we can immediately see that ϕ is a bijection. It is also clear that by combining nodes, we cannot increase the lengths of the existing cycles. Therefore, for any graph $G \in \mathcal{G}(n_k, \lambda, \rho)$, we have $\text{girth}(G) \geq \text{girth}(\phi(G))$. Therefore, we can say that

$$\begin{aligned} &\phi^{-1}\left(\mathcal{G}_g(n_k^*, x^{d-1}, x^{d-1})\right) \subseteq \mathcal{G}_g(n_k, \lambda, \rho) \\ &\Rightarrow \left|\mathcal{G}_g(n_k^*, x^{d-1}, x^{d-1})\right| \leq |\mathcal{G}_g(n_k, \lambda, \rho)| \end{aligned} \tag{11}$$

Using (10) and (11), we obtain

$$\frac{|\mathcal{G}_g(n_k, \lambda, \rho)|}{|\mathcal{G}(n_k, \lambda, \rho)|} \geq \frac{\left|\mathcal{G}_g(n_k^*, x^{d-1}, x^{d-1})\right|}{\left|\mathcal{G}(n_k^*, x^{d-1}, x^{d-1})\right|}$$

By Corollary 4.8, the right-hand-side term in the above inequality is positive for large enough n_k^*. Therefore, we have

$$\frac{|\mathcal{G}_g(n_k, \lambda, \rho)|}{|\mathcal{G}(n_k, \lambda, \rho)|} > 0$$

for large enough k. □

4.4 Asymptotic Block-Error Probability

Theorem 4.10. *Let (λ, ρ) be a DDP with minimum variable node degree $\mathtt{l}_{\min}$, maximum variable node degree $\mathtt{l}_{\max}$ and maximum check node degree $\mathtt{r}_{\max} > 2$. If the code $\mathcal{C}_1$ is chosen uniformly at random from $\mathcal{G}_{2k-2}(n, \lambda, \rho)$ and if $\varepsilon < \varepsilon_{\text{ef}}$, we have*

$$\mathbb{E}\left(P_B^{\text{BP}}(\mathcal{C}_1, \varepsilon)\right) = \mathcal{O}\left(\frac{1}{n^{\lceil \frac{\mathtt{l}_{\min}}{2} k \rceil - k}}\right)$$

and in the limits of small ε and large n,

$$\mathbb{E}\left(P_B^{\text{BP}}(\mathcal{C}_1, \varepsilon)\right) = \mathcal{O}\left(\frac{\varepsilon^k}{n^{\lceil \frac{\mathtt{l}_{\min}}{2} k \rceil - k}}\right)$$

4.4.1 Proof of Theorem 4.10

Let V_e be the set of variable nodes corresponding to the random erasures that occur during transmission. The BP decoder fails iff V_e contains a stopping set. So,

$$P_B^{\text{BP}}(\mathcal{C}_1, \varepsilon) = \Pr(\exists S \in \mathbb{S}(\mathcal{C}_1) : S \subset V_e)$$

For any $\delta_1, \delta_2 > 0$, we bound $P_B^{\text{BP}}(\mathcal{C}_1, \varepsilon)$ using union bound as

$$\begin{aligned} P_B^{\text{BP}}(\mathcal{C}_1, \varepsilon) \leq & \Pr\big(\exists S \in \mathbb{S}(\mathcal{C}_1) : S \subset V_e, k \leq |S| \leq \delta_1 n\big) \\ & + \Pr\big(\exists S \in \mathbb{S}(\mathcal{C}_1) : S \subset V_e, \delta_1 n \leq |S| \leq (\varepsilon + \delta_2) n\big) \\ & + \Pr\big(\exists S \in \mathbb{S}(\mathcal{C}_1) : S \subset V_e, (\varepsilon + \delta_2) n \leq |S| \leq n\big) \end{aligned}$$

The constants δ_1, δ_2 can be arbitrarily chosen by us and we will carefully set their values in the remainder of the proof. In the above, we only count stopping sets of size k or more because by restricting girth to $2k$ or more, we have eliminated stopping sets of size less than k. The probability $P_B^{\text{BP}}(\mathcal{C}_1, \varepsilon)$ is a function of the random variable

$\mathcal{C}_1$. The expected value of the function is

$$\begin{aligned}
\mathbb{E}\left(P_B^{\mathrm{BP}}(\mathcal{C}_1,\varepsilon)\right) &\le \mathbb{E}\Big(\Pr\big(\exists S \in \mathbb{S}(\mathcal{C}_1) : S \subset V_e, k \le |S| \le \delta_1 n\big)\Big) \\
&\quad + \mathbb{E}\Big(\Pr\big(\exists S \in \mathbb{S}(\mathcal{C}_1) : S \subset V_e, \delta_1 n \le |S| \le (\varepsilon+\delta_2)n\big)\Big) \\
&\quad + \mathbb{E}\Big(\Pr\big(\exists S \in \mathbb{S}(\mathcal{C}_1) : S \subset V_e, (\varepsilon+\delta_2)n \le |S| \le n\big)\Big) \\
\Rightarrow \mathbb{E}\left(P_B^{\mathrm{BP}}(\mathcal{C}_1,\varepsilon)\right) &\le \frac{1}{p}\mathbb{E}\Big(\Pr\big(\exists S \in \mathbb{S}(\mathcal{C}) : S \subset V_e, k \le |S| \le \delta_1 n\big)\Big) \\
&\quad + \frac{1}{p}\mathbb{E}\Big(\Pr\big(\exists S \in \mathbb{S}(\mathcal{C}) : S \subset V_e, \delta_1 n \le |S| \le (\varepsilon+\delta_2)n\big)\Big) \\
&\quad + \frac{1}{p}\mathbb{E}\Big(\Pr\big(\exists S \in \mathbb{S}(\mathcal{C}) : S \subset V_e, (\varepsilon+\delta_2)n \le |S| \le n\big)\Big) \qquad (12)
\end{aligned}$$

Here, $\mathcal{C}$ is a code selected from $\mathcal{G}(n,\lambda,\rho)$ uniformly at random and $p>0$ is some real number such that

$$\frac{|\mathcal{G}_{2k-2}(n,\lambda,\rho)|}{|\mathcal{G}(n,\lambda,\rho)|} \ge p$$

for large enough n (Lemma 4.9).

The proof of the theorem follows from the following three claims.

Claim 4.11. *As $n\to\infty$ and $\varepsilon\to 0$, the first term in* (12) *decays as*

$$\mathbb{E}\Big(\Pr\big(\exists S \in \mathbb{S}(\mathcal{C}) : S \subset V_e, k \le |S| \le \delta_1 n\big)\Big) = \mathcal{O}\left(\frac{\varepsilon^k}{n^{\lceil \frac{1}{2}\min k\rceil - k}}\right)$$

Claim 4.12. *As n increases, the second and the third terms in* (12) *decay exponentially in n.*

Claim 4.12 was proved by Orlitsky, et al. [39, Proof of Thm. 16] in a related context. The condition $\varepsilon < \varepsilon_{\mathrm{ef}}$ is required in their proof of this claim. The proof of Claim 4.11, provided below, is our novel contribution.

Proof of Claim 4.11. The probability that the erasure pattern V_e contains a stopping

set of size between k and $\delta_1 n$ can be upper bounded by the union bound as follows

$$\begin{aligned}\Pr\big(\exists S \in \mathbb{S}(\mathcal{C}) : S \subset V_e, k \le |S| \le \delta_1 n\big) &\le \sum_{i=k}^{\delta_1 n} \big|\{S \in \mathbb{S}(\mathcal{C}) : |S| = i\}\big| \Pr\left(S \in V_e\right) \\ &= \sum_{i=k}^{\delta_1 n} \big|\{S \in \mathbb{S}(\mathcal{C}) : |S| = i\}\big| \, \varepsilon^i \\ &= \sum_{i=k}^{\delta_1 n} \varepsilon^i \, \mathcal{E}(i) \end{aligned} \tag{13}$$

Now, we analyze $\mathcal{E}(i)$, the average number of stopping sets of size i in the standard ensemble. A stopping set of i variable nodes can have nodes of different degrees. Let us denote the number of degree-s variable nodes in the variable-node subset by i_s. Let $\mathcal{S}_i$ denote the set of all non-negative integer solutions to the equation $i_{\mathtt{l}_{\min}} + i_{\mathtt{l}_{\min}+1} + \cdots + i_{\mathtt{l}_{\max}} = i$. In other words, $\mathcal{S}_i$ is the number of ways we can select the degrees of the i variable nodes. The number of ways we can form a variable-node subset $S \subseteq V_v$ such that it has i_s variable nodes of degree s for all s is given by

$$\binom{nL_{\mathtt{l}_{\min}}}{i_{\mathtt{l}_{\min}}}\binom{nL_{\mathtt{l}_{\min}+1}}{i_{\mathtt{l}_{\min}+1}}\cdots\binom{nL_{\mathtt{l}_{\max}}}{i_{\mathtt{l}_{\max}}}$$

We can choose δ_1 to be sufficiently small so that i, which is always smaller than $\delta_1 n$, cannot exceed nL_{i_s} for any s. The probability that this variable node subset S is a stopping set is given by

$$\mathbb{E}\Pr\big(S \in \mathbb{S}(\mathcal{C})\big) = \frac{\text{no. of ways to connect } S \text{ to form a stopping set}}{\text{no. of ways to connect } S}$$

If we denote the number of ways to connect S to form a stopping set by A, we have

$$\mathbb{E}\Pr\big(S \in \mathbb{S}(\mathcal{C})\big) = \frac{A}{\binom{|E|}{\sum s i_s}}$$

By noting that

$$\binom{nL_{\mathtt{l}_{\min}}}{i_{\mathtt{l}_{\min}}}\binom{nL_{\mathtt{l}_{\min}+1}}{i_{\mathtt{l}_{\min}+1}}\cdots\binom{nL_{\mathtt{l}_{\max}}}{i_{\mathtt{l}_{\max}}} \le \binom{n}{i}$$

we can say that

$$\begin{aligned}\varepsilon^i \, \mathcal{E}(i) &= \varepsilon^i \sum_{\{i_s\} \in \mathcal{S}_i} \binom{nL_{\mathtt{l}_{\min}}}{i_{\mathtt{l}_{\min}}}\binom{nL_{\mathtt{l}_{\min}+1}}{i_{\mathtt{l}_{\min}+1}}\cdots\binom{nL_{\mathtt{l}_{\max}}}{i_{\mathtt{l}_{\max}}} \frac{A}{\binom{|E|}{\sum s i_s}} \\ &\le \varepsilon^i \binom{n}{i} \sum_{\{i_s\} \in \mathcal{S}_i} \frac{A}{\binom{|E|}{\sum s i_s}} \end{aligned} \tag{14}$$

The quantity A is the number of ways of choosing $\sum si_s$ check-node sockets such that we select at least two sockets from each participating check node. As long as i is a small fraction of n, which we can ensure by selecting a small enough δ_1, we can keep A independent of n (for a given i). Also note that if we increase the degree of all the check nodes, A can only increase. Therefore, we can upper bound A by the number of ways to select check-node sockets assuming that all check nodes have the maximum degree $\mathtt{r}_{\max}$.

For a non-negative integer $r_j \leq \mathtt{r}_{\max}$, the number of ways of choosing r_j sockets from a check node of degree $\mathtt{r}_{\max}$ is $\binom{\mathtt{r}_{\max}}{r_j}$. To form a stopping set, we must choose r_j sockets from the j^{th} check node in the Tanner graph such that $r_j \in \{0, 2, 3, 4, \ldots, \mathtt{r}_{\max}\}$ and $\sum_{j=1}^{m} r_j = \sum si_s$. The total number of ways to select the sockets (assuming maximum check node degree for all check nodes) is

$$\sum_{\substack{(r_j):\sum_{j=1}^{m} r_j = \sum si_s \\ r_j \in \{0,2,3,\ldots,\mathtt{r}_{\max}\},\forall j}} \prod_{j=1}^{m} \binom{\mathtt{r}_{\max}}{r_j} = \operatorname{coef}\Big(\big((1+x)^{\mathtt{r}_{\max}} - \mathtt{r}_{\max} x\big)^m, x^{\sum si_s}\Big)$$

Here, $\operatorname{coef}\big(p(x), x^r\big)$ denotes the coefficient of x^r in the polynomial $p(x)$. Note that we are subtracting the term $\mathtt{r}_{\max} x$ because $r_j \neq 1$ for any j. We can now upper bound the quantity A as follows.

$$\begin{aligned} A &\leq \operatorname{coef}\left(\big((1+x)^{\mathtt{r}_{\max}} - \mathtt{r}_{\max} x\big)^m, x^{\sum si_s}\right) \\ &\leq \binom{m + \lfloor \frac{\sum si_s}{2} \rfloor - \lceil \frac{\sum si_s}{\mathtt{r}_{\max}} \rceil}{\lfloor \frac{\sum si_s}{2} \rfloor} (2\mathtt{r}_{\max} - 3)^{\sum si_s} \end{aligned}$$

where the last inequality follows from [39, Lemma 18]. If we denote $\sum si_s$ by w, we

have $i\mathtt{l}_{\min} \le w \le i\mathtt{l}_{\max}$. Substituting the above in (14), we get

$$\begin{aligned}
\varepsilon^i \mathcal{E}(i) &\le \varepsilon^i \binom{n}{i} \sum_{\{i_s\}\in\mathcal{S}_i} \binom{m + \lfloor \frac{w}{2} \rfloor - \lceil \frac{w}{\mathtt{r}_{\max}} \rceil}{\lfloor \frac{w}{2} \rfloor} \frac{(2\mathtt{r}_{\max}-3)^w}{\binom{|E|}{w}} \\
&\le \varepsilon^i \binom{n}{i} (2\mathtt{r}_{\max}-3)^{i\mathtt{l}_{\max}} \sum_{\{i_s\}\in\mathcal{S}_i} \binom{m + \lfloor \frac{w}{2} \rfloor - \lceil \frac{w}{\mathtt{r}_{\max}} \rceil}{\lfloor \frac{w}{2} \rfloor} \frac{1}{\binom{|E|}{w}} \\
&\le \varepsilon^i \binom{n}{i} (2\mathtt{r}_{\max}-3)^{i\mathtt{l}_{\max}} \sum_{\{i_s\}\in\mathcal{S}_i} \binom{m + \frac{i\mathtt{l}_{\max}}{2}}{\lfloor \frac{w}{2} \rfloor} \frac{1}{\binom{|E|}{w}} \\
&\le \varepsilon^i \binom{n}{i} (2\mathtt{r}_{\max}-3)^{i\mathtt{l}_{\max}} \sum_{\{i_s\}\in\mathcal{S}_i} \frac{\left(m + \frac{i\mathtt{l}_{\max}}{2}\right)^{\lfloor \frac{w}{2} \rfloor} w!}{\lfloor \frac{w}{2} \rfloor! \left(|E| - i\mathtt{l}_{\max}\right)^w} \qquad (15)
\end{aligned}$$

Note that in the last step we have used the following two inequalities.

$$\begin{aligned}
\binom{m + \frac{i\mathtt{l}_{\max}}{2}}{\lfloor \frac{w}{2} \rfloor} &\le \frac{\left(m + \frac{i\mathtt{l}_{\max}}{2}\right)^{\lfloor \frac{w}{2} \rfloor}}{\lfloor \frac{w}{2} \rfloor!} \\
\binom{|E|}{w} &\ge \frac{(|E|-w)^w}{w!} \ge \frac{(|E| - i\mathtt{l}_{\max})^w}{w!}
\end{aligned}$$

If we denote the summand in (15) by $f(w)$, we have

$$\frac{f(2r+1)}{f(2r)} = \frac{2r+1}{|E| - i\mathtt{l}_{\max}} \le \frac{i\mathtt{l}_{\max}}{|E| - i\mathtt{l}_{\max}} \le \frac{\delta_1 n \mathtt{l}_{\max}}{|E| - \delta_1 n \mathtt{l}_{\max}} \le 1$$

if we choose δ_1 small enough. Also,

$$\frac{f(2r+2)}{f(2r+1)} = 2\frac{m + \frac{i\mathtt{l}_{\max}}{2}}{|E| - i\mathtt{l}_{\max}} \le 2\frac{m + \frac{\delta_1 n \mathtt{l}_{\max}}{2}}{|E| - \delta_1 n \mathtt{l}_{\max}}$$

Since $\mathtt{r}_{\max} > 2$ we have $|E| > 2m$. Again, if we choose δ_1 small enough, we will have $f(2r+2)/f(2r+1) \le 1$. So, $f(w)$ is a non-increasing function and $w \ge i\mathtt{l}_{\min}$, we can upper bound all the summands in (15) by $f(i\mathtt{l}_{\min})$. We now have

$$\begin{aligned}
\varepsilon^i \mathcal{E}(i) &\le \varepsilon^i \binom{n}{i} (2\mathtt{r}_{\max}-3)^{i\mathtt{l}_{\max}} \times \sum_{\{i_s\}\in\mathcal{S}_i} \frac{\left(m + \frac{i\mathtt{l}_{\max}}{2}\right)^{\lfloor \frac{i\mathtt{l}_{\min}}{2} \rfloor} (i\mathtt{l}_{\min})!}{\lfloor \frac{i\mathtt{l}_{\min}}{2} \rfloor! \left(|E| - i\mathtt{l}_{\max}\right)^{i\mathtt{l}_{\min}}} \\
&\overset{\text{a}}{\le} \varepsilon^i \binom{n}{i} (2\mathtt{r}_{\max}-3)^{i\mathtt{l}_{\max}} (i+1)^{\mathtt{l}_{\max}} \times \frac{\left(m + \frac{\delta_1 n \mathtt{l}_{\max}}{2}\right)^{\lfloor \frac{i\mathtt{l}_{\min}}{2} \rfloor} (i\mathtt{l}_{\min})!}{\lfloor \frac{i\mathtt{l}_{\min}}{2} \rfloor! \left(|E| - \delta_1 n \mathtt{l}_{\max}\right)^{i\mathtt{l}_{\min}}} \\
&\le \varepsilon^i \binom{n}{i} (2\mathtt{r}_{\max}-3)^{i\mathtt{l}_{\max}} \frac{(i+1)^{\mathtt{l}_{\max}}}{n^{\lceil \frac{i\mathtt{l}_{\min}}{2} \rceil}} \times \frac{\left(\alpha + \frac{\delta_1 \mathtt{l}_{\max}}{2}\right)^{\lfloor \frac{i\mathtt{l}_{\min}}{2} \rfloor} (i\mathtt{l}_{\min})!}{\lfloor \frac{i\mathtt{l}_{\min}}{2} \rfloor! \left(\beta - \delta_1 \mathtt{l}_{\max}\right)^{i\mathtt{l}_{\min}}} \\
&\triangleq \varepsilon^i J_i \qquad (16)
\end{aligned}$$

In step (a), we have used the fact that $|\mathcal{S}_i| \le (i+1)^{\mathtt{l}_{\max}}$. Also, $\alpha \triangleq m/n$ and $\beta \triangleq |E|/n$ depend only on ρ and λ. If i remains constant as $n \to \infty$, we have

$$J_i = \Theta\left(\frac{1}{n^{\lceil \frac{i\mathtt{l}_{\min}}{2} \rceil - i}}\right)$$

Also,

$$\begin{aligned}
\frac{J_{i+2}}{J_i} &= \frac{\binom{n}{i+2}}{\binom{n}{i}} (2\mathtt{r}_{\max} - 3)^{2\mathtt{l}_{\max}} \frac{(\alpha + \frac{\delta_1 \mathtt{l}_{\max}}{2})^{\mathtt{l}_{\min}}}{(\beta - \delta_1 \mathtt{l}_{\max})^{2\mathtt{l}_{\min}}} \\
&\qquad \times \left(\frac{i+3}{i+1}\right)^{\mathtt{r}_{\max}} \frac{(i\mathtt{l}_{\min} + 2\mathtt{l}_{\min})! \lfloor \frac{i\mathtt{l}_{\min}}{2} \rfloor !}{(i\mathtt{l}_{\min})! \left(\lfloor \frac{i\mathtt{l}_{\min}}{2} \rfloor + \mathtt{l}_{\min}\right)!\ n^{\mathtt{l}_{\min}}} \\
&\le \frac{(n-i-1)(n-i)}{(i+1)(i+2)} (2\mathtt{r}_{\max} - 3)^{2\mathtt{l}_{\max}} \left(\frac{i+3}{i+1}\right)^{\mathtt{r}_{\max}} \\
&\qquad \times \frac{(\alpha + \frac{\delta_1 \mathtt{l}_{\max}}{2})^{\mathtt{l}_{\min}}}{(\beta - \delta_1 \mathtt{l}_{\max})^{2\mathtt{l}_{\min}}} \ \frac{(i\mathtt{l}_{\min} + 2\mathtt{l}_{\min})^{2\mathtt{l}_{\min}}}{\left(\lfloor \frac{i\mathtt{l}_{\min}}{2} \rfloor + 1\right)^{\mathtt{l}_{\min}} n^{\mathtt{l}_{\min}}}
\end{aligned}$$

Using $\frac{i+3}{i+1} \le 2$, $i\mathtt{l}_{\min} + 2\mathtt{l}_{\min} \le 3i\mathtt{l}_{\min}$, $\lfloor x \rfloor + 1 \ge x$,

$$\frac{J_{i+2}}{J_i} \le \frac{n^2}{i^2} (2\mathtt{r}_{\max} - 3)^{2\mathtt{l}_{\max}} 2^{\mathtt{r}_{\max}} \frac{(\alpha + \frac{\delta_1 \mathtt{l}_{\max}}{2})^{\mathtt{l}_{\min}}}{(\beta - \delta_1 \mathtt{l}_{\max})^{2\mathtt{l}_{\min}}} \times \frac{(3i\mathtt{l}_{\min})^{2\mathtt{l}_{\min}}}{\left(\frac{i\mathtt{l}_{\min}}{2}\right)^{\mathtt{l}_{\min}} n^{\mathtt{l}_{\min}}}$$

Choosing $\delta_3 \in (0,1)$ such that $\beta - \delta_3 \mathtt{l}_{\max} > 0$ and letting $\delta_1 < \delta_3$,

$$\begin{aligned}
\frac{J_{i+2}}{J_i} &\le (2\mathtt{r}_{\max} - 3)^{2\mathtt{l}_{\max}}\ 2^{\mathtt{r}_{\max}} \frac{(\alpha + \frac{\delta_3 \mathtt{l}_{\max}}{2})^{\mathtt{l}_{\min}} (3\mathtt{l}_{\min})^{2\mathtt{l}_{\min}}}{(\beta - \delta_3 \mathtt{l}_{\max})^{2\mathtt{l}_{\min}} \left(\frac{\mathtt{l}_{\min}}{2}\right)^{\mathtt{l}_{\min}}} \left(\frac{i}{n}\right)^{\mathtt{l}_{\min}-2} \\
&\triangleq B \left(\frac{i}{n}\right)^{\mathtt{l}_{\min}-2} \quad \le \quad B\delta_1^{\mathtt{l}_{\min}-2}
\end{aligned} \tag{17}$$

where B depends only on λ and ρ. Going back to (13), and substituting (16) and (17), we get

$$\begin{aligned}
\Pr\big(\exists S \in \mathbb{S}(\mathcal{C}) : S \subset V_e, k \le |S| \le \delta_1 n\big) &\le \sum_{i=k}^{\delta_1 n} \varepsilon^i\, \mathcal{E}(i) \le \sum_{i=k}^{\delta_1 n} \varepsilon^i J_i \le \varepsilon^k \sum_{i=k}^{\delta_1 n} J_i \\
&\le \varepsilon^k \left(J_k + J_{k+1}\right) \sum_{i=0}^{\lceil \delta_1 n/2 \rceil} \left(B\delta_1^{\mathtt{l}_{\min}-2}\right)^i
\end{aligned}$$

If δ_1 is small enough, then the summation in the final expression is bounded by a decreasing geometric sum. Moreover,

$$J_k = \Theta\left(\frac{1}{n^{\lceil \frac{\mathtt{l}_{\min}}{2} k \rceil - k}}\right), \quad J_{k+1} = \Theta\left(\frac{1}{n^{\lceil \frac{\mathtt{l}_{\min}}{2}(k+1) \rceil - k - 1}}\right)$$

and among these two terms, J_k has a slower decay. Therefore,

$$\Pr\left(\exists S \in \mathbb{S}(\mathcal{C}) : S \subset V_e, k \leq |S| \leq \delta_1 n\right) = \mathcal{O}\left(\frac{\varepsilon^k}{n^{\lceil \frac{\mathtt{l}_{\min}}{2} k \rceil - k}}\right)$$

as $\varepsilon \to 0$ and $n \to \infty$. □

4.5 Secrecy Regions

Recall that for strong secrecy on BEWC(ξ), we require a sequence of codes whose block-error probability over BEC($1-\xi$) decays as $1/n^2$. From Theorem 4.10, we know that the block-error probability of LDPC codes without cycles of length $2k-2$ or smaller, and minimum left degree $\mathtt{l}_{\min}$ decays as

$$\mathcal{O}\left(\frac{1}{n^{\lceil \frac{\mathtt{l}_{\min}}{2} k \rceil - k}}\right)$$

over BEC(ε) for $\varepsilon < \varepsilon_{\text{ef}}$. In particular, the duals of (λ, ρ) LDPC codes with $\mathtt{l}_{\min} \geq 3$ and without cycles of length four or less will achieve $\mathcal{O}(1/n^2)$ block-error *on an average*. Loosely speaking, a randomly selected code from this ensemble can be "expected" to achieve strong secrecy on BEWC(ξ) for all $\xi > 1 - \varepsilon_{\text{ef}}$. In a stricter sense, this is an existence result—there exists an LDPC code in the aforementioned ensemble that achieves strong secrecy for all $\xi > 1 - \varepsilon_{\text{ef}}$.

4.5.1 The Value of ε_{ef}

Given the value of the function $\gamma(\alpha)$ for $\alpha \in [0,1]$, the computation of ε_{ef} is straightforward. The computation of $\gamma(\alpha)$ depends on whether the the DDP corresponds to a regular or an irregular LDPC code.

Theorem 4.13 ([39, Thm. 2]). *For $(\mathtt{l}, \mathtt{r})$ regular LDPC codes, we have*

$$\gamma(\alpha) = \frac{\mathtt{l}}{\mathtt{d}} \log_2\left(\frac{(1+x_0)^{\mathtt{r}} - \mathtt{r} x_0}{x_0^{\alpha \mathtt{r}}}\right) - (\mathtt{l}-1) h(\alpha)$$

where x_0 is the only positive solution of

$$\frac{x\left((1+x)^{\mathtt{r}-1} - 1\right)}{(1+x)^{\mathtt{r}} - \mathtt{r} x} = \alpha$$

For an irregular DDP, the computation of ε_{ef} is much more involved and we refer the reader to [39, Lemma 4, Thm. 5].

Since the computation of ε_{ef} is completely indirect, there is no known direct analytic technique to track its value. For our strong secrecy application, we are interested in achieving large values of ε_{ef}. The following upper bound is obvious

$$\varepsilon_{\text{ef}} \leq \varepsilon_{\text{th}} < 1 - R \tag{18}$$

The closer bound follows from two facts.

1. For $\varepsilon < \varepsilon_{\text{ef}}$, the probability of block-error for a randomly selected LDPC code from the standard ensemble over BEC(ε) goes to zero [39, Thm. 16].

2. For $\varepsilon > \varepsilon_{\text{th}}$, the BER of a randomly selected LDPC code from the standard ensemble over BEC(ε) is bounded away from zero.

We performed numerical computation of ε_{ef} for a collection of DDPs and found that ε_{ef} is bounded away from ε_{th}. Fig. 18 shows a plot of the values for rate-$\frac{1}{2}$ DDPs optimized for high ε_{th} using the LDPCOPT online database [42]. In this case, we also note that the maximum value of ε_{ef} is achieved by the $(3, 6)$ regular LDPC code.

4.5.2 Strong and Weak Secrecy Regions

In this chapter, we consider a DDP of rate $\mathtt{R}$ and $\mathtt{l}_{\min} \geq 3$, analyze the strong secrecy region—the values of ξ for which there exist LDPC codes with girth at least six achieve strong secrecy—for the BEWC(ξ). We have shown that

$$1 - \varepsilon_{\text{ef}} < \xi \leq 1$$

Thangaraj, et al. [1, §IV-B] considered achieving weak secrecy using the duals of LDPC codes over the BEWC and showed that the weak secrecy region is given by

$$1 - \varepsilon_{\text{th}} < \xi \leq 1$$

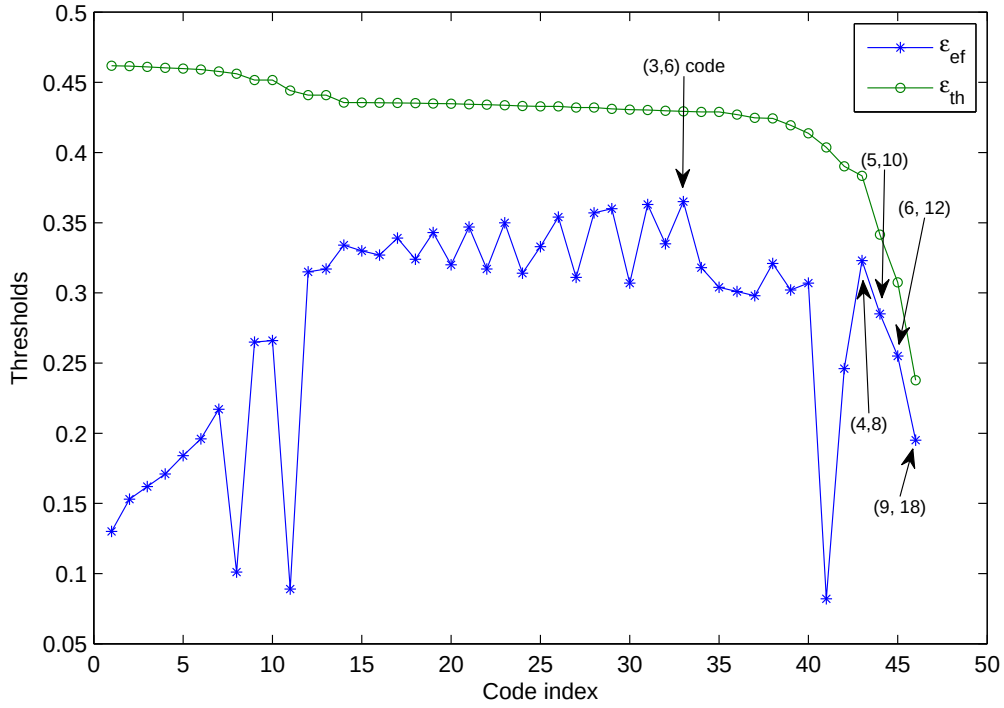


Figure 18: The values of $\varepsilon_{\mathrm{ef}}$ and $\varepsilon_{\mathrm{th}}$ for threshold-optimized rate-$\frac{1}{2}$ LDPC codes. The arrows indicate the values for regular LDPC codes.

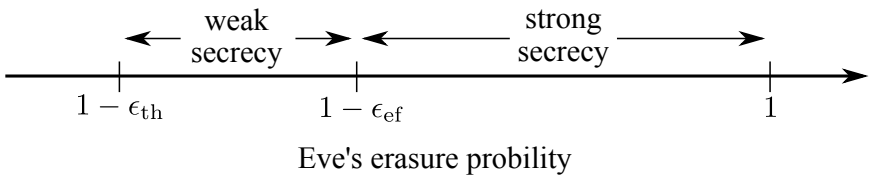


Figure 19: A sketch of the weak and strong secrecy regions achieved by short-cycle-free LDPC codes.

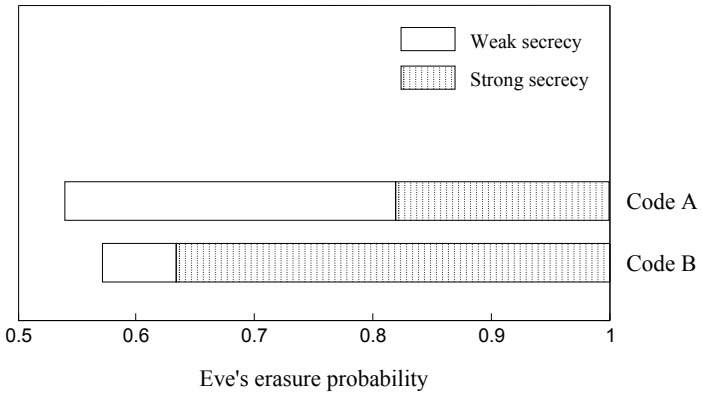


Figure 20: Weak and strong secrecy regions achieved by the DDPs $(0.9131x^2 + 0.0124x^{17} + 0.0651x^{18} + 0.009363x^{70}, 0.2703x^8 + 0.7297x^9)$ (Code A) and (x^2, x^5) (Code B).

Combining the above two secrecy regions and using the bound (18), we sketch the secrecy region of short-cycle-free LDPC codes in Fig. 19.

For example, Fig. 20 shows the secrecy regions of two rate-$\frac{1}{2}$ LDPC codes. Code A is a threshold-optimized LDPC code and Code B is the $(3, 6)$ regular LDPC code. Code A has $\varepsilon_{\text{th}} \approx 0.460$ and $\varepsilon_{\text{ef}} \approx 0.184$ and the duals will achieve weak secrecy for $\xi \in (0.540, 0.816]$ and strong secrecy for $\xi \in (0.816, 1]$. The $(3, 6)$ regular code has $\varepsilon_{\text{th}} \approx 0.429$ and $\varepsilon_{\text{ef}} \approx 0.365$. Therefore, the dual codes will achieve weak secrecy for $\xi \in (0.571, 0.635]$ and strong secrecy for $\xi \in (0.635, 1]$.

CHAPTER V

LARGE-GIRTH LDPC CODES FOR STRONG SECRECY

In Chapter 3, we showed that we can achieve strong secrecy on the BEWC model using the duals of codes whose block-error probability decays as $1/n^2$ as the block length n increases. A solution to this problem was discussed in Chapter 4, where we considered short-cycle-free LDPC codes, i.e., LDPC codes whose Tanner graphs did not have any cycles of length less than six. Using a stopping-set analysis, we studied the asymptotic block-error probability of these codes under BP decoding and showed that they indeed satisfy the requirement for strong secrecy.

In this chapter, we consider "large-girth" LDPC codes, i.e., LDPC codes corresponding to Tanner graphs whose girth grows logarithmically fast as the block length increases. Instead of directly studying the block-error rate of these codes, we study their bit-error rate using density-evolution analysis and use it to upper bound the block-error rate. For certain large-girth LDPC codes, we show that the BER and the block-error rate over BEC decay in a sub-exponential manner, i.e., as $\mathcal{O}\left(\exp\left(-c_1 n^{c_2}\right)\right)$ for some constants $c_1 > 0$ and $0 < c_2 \leq 1$, when the erasure probability is below the threshold. The results pertaining to this chapter were published in [43].

This chapter is organized as follows. We begin with a discussion of the density evolution analysis of LDPC codes over the BEC. In the second section, we show that the density evolution BER estimate decays in a double-exponential manner below the BEC threshold as the number of iterations increases. In the third section, we motivate the use of large-girth LDPC codes for strong secrecy and provide a short survey of some prior work regarding LDPC codes with high girth. In the fourth section, we provide a brief overview of Lubotzky-Phillips-Sarnak (LPS) graphs, their

properties, and their applications in error-correction coding. In the fifth section, we provide an algorithm to construct large-girth LDPC codes from LPS graphs. The sub-exponential fall of the BER of these LDPC codes is proved in the sixth section. In the final section, we discuss the strong secrecy region achieved by our large-girth LDPC codes.

5.1 Background on Density Evolution

As discussed in our brief introduction to LDPC codes (§4.1) in the previous chapter, the performance of individual LDPC codes is not usually studied; instead, the analysis focuses on the average performance of an ensemble of LDPC codes. One of the quantities of interest is the average BER of LDPC codes uniformly sampled from the standard ensemble. One of the ways to study this quantity is to use an analysis technique called *density evolution*. Density evolution provides a generalized framework to analyze the bit-error rate of sparse-graph codes over symmetric channels. The analysis in this chapter is limited to LDPC codes over the BEC. In the following discussion, we will give a brief introduction to the density evolution analysis of the bit-error rate of the BP decoder when it decodes an LDPC codeword transmitted over a BEC. The material in this section is based on the discussion in the book *Modern Coding Theory* [34, §3.7-3.11].

Let $\mathcal{H}_n$ be an arbitrary ensemble of Tanner graphs with n variable nodes. Suppose a graph G is selected uniformly at random from $\mathcal{H}_n$ and a random codeword from the associated code is transmitted over BEC(ε). The receiver, with the knowledge of G, tries to decode the transmitted word using the BP decoding algorithm. For a family of ensembles $(\mathcal{H}_n)$ with increasing block length n, let

- $x(t, n)$ = the probability that a randomly selected edge in the Tanner graph G transmits an erasure message from its variable node to its check node at the t^{th} iteration

- $y(t,n)$ = the probability that a randomly selected codeword bit is unknown after t iterations. In other words, $y(t,n)$ is the average bit-error probability of $\mathcal{H}_n$ after t iterations of BP decoding.

5.1.1 Computation Graphs

To evaluate $x(t,n)$ and $y(t,n)$ explicitly, the *computation graphs* [34, §3.7] associated with the Tanner graph ensemble $\mathcal{H}_n$ may be considered. Suppose a graph G is selected from $\mathcal{H}_n$ uniformly at random and a random edge e is picked from G. Let v be the variable node connected to e. The level-t *edge-rooted computation graph* $\vec{\mathcal{C}}_t$ of $\mathcal{H}_n$ is defined as the subgraph obtained by traversing from v up to iteration depth t in all initial directions except along e. Since the selection of the graph G and the edge e are both random, the computation graph $\vec{\mathcal{C}}_t$ is a random graph whose distribution depends only on t and $\mathcal{H}_n$. Also, $x(t,n)$ can be uniquely determined given the possible realizations of $\vec{\mathcal{C}}_t$ and their probabilities.

To evaluate $y(t,n)$, the level-t *node-rooted computation graph* $\mathring{\mathcal{C}}_t$, defined subsequently, may be considered. As before, a graph G is selected from $\mathcal{H}_n$ uniformly at random. Then, a variable node v is picked from G uniformly at random. The graph $\mathring{\mathcal{C}}_t$ is defined as the subgraph obtained by traversing from v up to iteration depth t in *all* directions. Like $\vec{\mathcal{C}}_t$, the distribution of $\mathring{\mathcal{C}}_t$ is also dependent only on t and $\mathcal{H}_n$. The probability $y(t,n)$ can be uniquely determined given the possible realizations of $\mathring{\mathcal{C}}_t$ and their probabilities.

Fig. 21(a) shows the level-2 edge-rooted computation graph corresponding to the edge e in the Tanner graph in Fig. 21(b). Fig. 22(b) shows the level-2 node-rooted computation graph corresponding to the variable node v_1 in the same Tanner graph. Note that in both the figures, we have duplicated some of the nodes and we have drawn the graphs like a tree. This is done for the sake of simplicity in visualization. The actual computation graphs are constructed by identifying duplicate vertices and

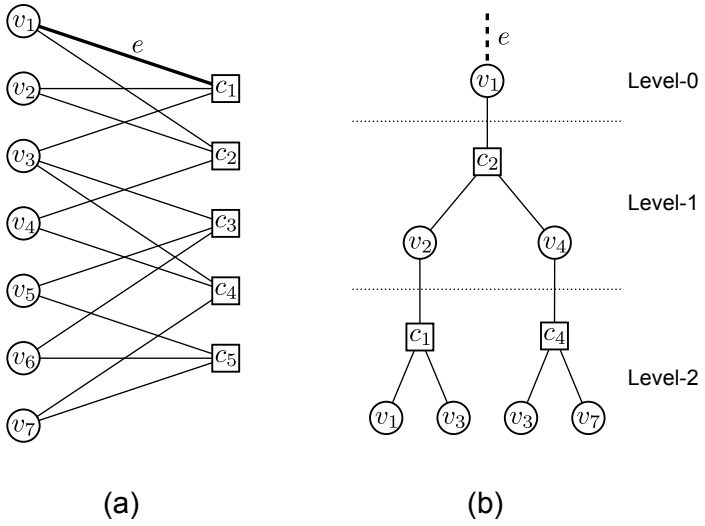


Figure 21: A Tanner graph (a) and the level-2 computation graph (b) rooted at the edge e. The edge e (dotted) is not a part of the computation graph.

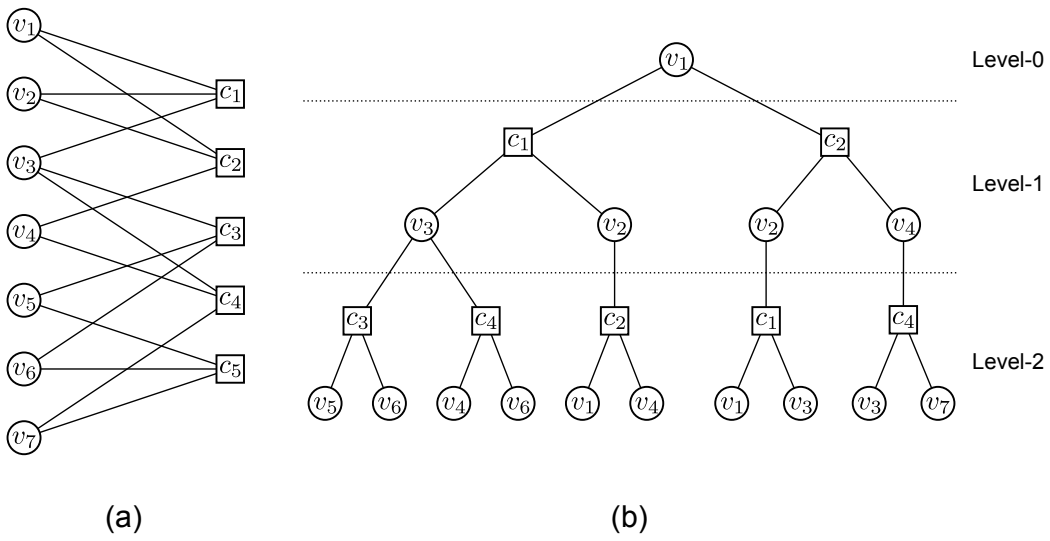


Figure 22: A Tanner graph (a) and the level-2 computation graph (b) rooted at the node v_1.

duplicate edges in the two figures, and they will contain some cycles.

It can be noted that even though the computation graphs in Figs. 21 and 22 are expanded from the same variable node, they are different. In general, the computation graphs $\vec{\mathcal{C}}_t$ and $\mathring{\mathcal{C}}_t$ are slightly different in the following two aspects.

- The root node degrees are different in the two graphs. Loosely speaking, the root node degree in $\mathring{\mathcal{C}}_t$ is one more than that of $\vec{\mathcal{C}}_t$. This is because, while constructing $\vec{\mathcal{C}}_t$, we don't travel along the root edge.

- Even ignoring the difference in the degrees, the degree distribution of the "root" variable node is different for $\vec{\mathcal{C}}_t$ and $\mathring{\mathcal{C}}_t$. This is because, the degree of a variable node attached to a randomly chosen edge is dictated by $\lambda(x)$, the edge-perspective degree distribution, while that of a randomly chosen variable node is dictated by $L(x)$, the node-perspective degree distribution.

5.1.2 Tree Ensembles

While studying the error-correcting performance of LDPC codes, the codes corresponding to the standard ensemble of Tanner graphs $\mathcal{G}(n, \lambda, \rho)$ are usually considered. The graphs in this ensemble contain n variable nodes, whose degrees are determined by the degree distribution polynomial $\lambda(x) = \sum_i \lambda_i x^{i-1}$, where λ_i is the fraction of edges that are connected to degree-i variable nodes. The check-node degree distribution is determined by the polynomial $\rho(x) = \sum_j \rho_j x^{j-1}$, where ρ_j is the fraction of edges connected to degree-j check nodes.

In the classical setting, $\mathcal{H}_n = \mathcal{G}(n, \lambda, \rho)$ is considered, and $x(t, n)$ and $y(t, n)$ are analyzed while keeping t fixed and letting n grow monotonically. The possible computation graphs of $\mathcal{G}(n, \lambda, \rho)$ are not cycle-free and hence enumerating them is cumbersome. Doing an exact analysis of the average bit-error probability $y(t, n)$ for this ensemble is therefore difficult. By enumerating the computation graphs, one can note that for a fixed iteration depth t and increasing n the computation graphs $\vec{\mathcal{C}}_t$

and $\mathring{\mathcal{C}}_t$ will be trees with probability converging to unity at a rate that is $\mathcal{O}(1/n)$. In other words,

$$\Pr\left(\vec{\mathcal{C}}_t \text{ is not a tree}\right), \Pr\left(\mathring{\mathcal{C}}_t \text{ is not a tree}\right) = \mathcal{O}\left(\frac{1}{n}\right)$$

The term *density evolution* refers to the calculation of the bit-error probability by assuming that the computation graphs are identical to the limiting trees. The density evolution analysis is so called because it analyzes the evolution of probability density of the messages sent along the edges as the number of decoding iterations increases.

The *level-t node-rooted tree ensemble* $\mathring{\mathcal{T}}_t$ corresponding to the DDP (λ, ρ) is a random tree that has a Tanner-graph-like bipartite structure. The tree $\mathring{\mathcal{T}}_t$ is built iteratively starting from a root variable node according to the following steps.

1. The root variable node of the tree has i sockets with probability L_i, where L_i is the fraction of degree-i variable nodes allowed by the DDP (λ, ρ).

2. For levels 1 to t, do the following

   (a) For each free variable-node socket in the previous level, add a check node. Each such check node has j sockets with probability ρ_j and one of these sockets connects to the corresponding variable-node socket in the previous level. The number of sockets in a check node is independent from all other events.

   (b) For each free check-node socket from the previous step, add a variable node. A variable-node added at this way will have i sockets with probability λ_i and one of these sockets connects to the corresponding check-node socket. The number of sockets in the variable nodes are chosen independently.

In the above process, the leaf variable nodes, which are the level-t variable nodes, will have only one fulfilled socket. The unfulfilled sockets are discarded from the construction.

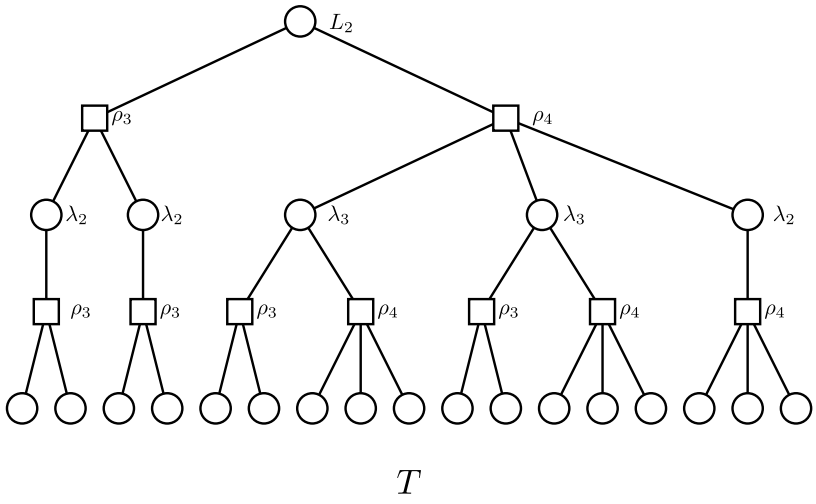


Figure 23: An instance T of the node-rooted tree ensemble and the probabilities associated with the node degrees.

The *level-t edge-rooted tree ensemble* $\vec{\mathcal{T}}_t$ is constructed with a nearly identical algorithm except for the fact that the root variable node has degree i with probability λ_{i+1}. It can be shown that $\vec{\mathcal{T}}_t$ is an asymptotic approximation of $\vec{\mathcal{C}}_t$, and $\mathring{\mathcal{T}}_t$ is an asymptotic approximation of $\mathring{\mathcal{C}}_t$.

For a given tree T, the probability of the tree ensemble $\mathring{\mathcal{T}}_t$ being equal to T can be calculated as follows. Suppose the tree T has q_j check nodes of degree j and p_i non-leaf variable nodes of degree i, with the root variable node having degree i_0. Then,

$$\Pr\left(\mathring{\mathcal{T}}_t = T\right) = L_{i_0} \, \frac{1}{\lambda_{i_0}} \prod_{i=\mathtt{l}_{\min}}^{\mathtt{l}_{\max}} \lambda_i^{p_i} \prod_{j=\mathtt{r}_{\min}}^{\mathtt{r}_{\max}} \rho_j^{q_j}$$

The probability of the edge-rooted tree ensemble $\vec{\mathcal{T}}_t$ being equal to T is given by

$$\Pr\left(\vec{\mathcal{T}}_t = T\right) = \lambda_{i_0+1} \, \frac{1}{\lambda_{i_0}} \prod_{i=\mathtt{l}_{\min}}^{\mathtt{l}_{\max}} \lambda_i^{p_i} \prod_{j=\mathtt{r}_{\min}}^{\mathtt{r}_{\max}} \rho_j^{q_j}$$

For example, given the DDP (λ, ρ), the probability of the node-rooted tree ensemble $\mathring{\mathcal{T}}_2$ is equal to the tree T in Fig. 23 is given by the product of all the node-degree probabilities in the figure. That is,

$$\Pr\left(\mathring{\mathcal{T}}_t = T\right) = L_2 \, \rho_3^5 \, \rho_4^4 \, \lambda_2^3 \, \lambda_3^2$$

5.1.3 The Density Evolution Equations

The tree ensemble $\vec{\mathcal{T}}_t$ is a Tanner graph, and hence it has a binary linear code associated with it. Suppose a random codeword from this code is transmitted over BEC(ε) and decoded using t iterations of the BP decoding algorithm. Let x_t denote the bit-error probability of the root variable node after BP decoding. A similar probability y_t for the tree ensemble $\mathring{\mathcal{T}}_t$ can be defined. It can be easily seen that x_t and y_t are given by the following recursive relationship.

$$\begin{aligned} x_0 &= \varepsilon \\ x_t &= \varepsilon\lambda(1-\rho(1-x_{t-1})), \qquad t>0 \\ y_t &= \varepsilon L(1-\rho(1-x_{t-1})), \qquad t>0 \end{aligned}$$

The above equations are the density evolution equations for the BEC.

Definition 5.1 (Threshold)**.** The erasure threshold ε_{th} of an LDPC DDP (λ,ρ) is defined as the supremum of the erasure parameter ε for which the density evolution estimate for BEC x_t, or equivalently y_t, converges to zero. That is,

$$\varepsilon_{\text{th}} \triangleq \sup\left\{\varepsilon>0 : x_t \overset{t\to\infty}{\longrightarrow} 0\right\}$$

▼

In other words, the threshold is loosely defined as the "worst" channel parameter for which the density evolution BER estimate converges to zero.

5.1.4 Density Evolution Estimate vs. Bit-Error Probability

The quantity y_t given by density evolution is an *estimate* of the average bit-error probability when a random code $\mathcal{C}_n$ from the standard ensemble $\mathcal{G}(n,\lambda,\rho)$ of LDPC codes is transmitted over BEC(ε). For a given code $\mathcal{C}_n$, let $P_b^{\text{BP}}(\mathcal{C}_n,\varepsilon,t)$ denote the probability of bit error after t iterations of BP decoding. It can be shown that the

average bit-error probability $\mathbb{E}\left(P_b^{\mathrm{BP}}(\mathcal{C}_n, \varepsilon, t)\right) \triangleq y(t, n)$ approaches y_t as long as t remains constant and n increases. In particular, it can be shown that for a constant integer t

$$\left|\mathbb{E}\left(P_b^{\mathrm{BP}}(\mathcal{C}_n, \varepsilon, t)\right) - y_t\right| = \mathcal{O}\left(\frac{1}{n}\right), \quad \text{as } n \to \infty$$

5.2 Asymptotic Behavior of Density Evolution Estimate

It is a well-known result that the quantities x_t and y_t exhibit a double-exponential decay as t goes to infinity for $\varepsilon < \varepsilon_{\mathrm{th}}$. A proof of this result for regular codes was provided by Lentmaier, et al. [44, §V-A]. For the sake of completeness, we state the more general result for irregular codes and provide an alternative proof involving mathematical induction.

Lemma 5.2. *For a DDP (λ, ρ) with minimum variable node degree $\mathtt{l}_{\min} \geq 3$ and $\varepsilon < \varepsilon_{\mathrm{th}}$, we have*

$$x_t, y_t = \mathcal{O}\left(\exp(-\beta(\mathtt{l}_{\min} - 1)^t)\right) \tag{19}$$

as $t \to \infty$, where $\beta > 0$ is a constant.

Proof. For any $x \in [0, 1]$, we have

$$\begin{aligned}
(1-x)^{d-1} &\geq 1 - (d-1)x, \qquad \forall d \in \mathbb{N} \\
\Rightarrow \rho(1-x) &= \sum_{d=\mathtt{r}_{\min}}^{\mathtt{r}_{\max}} \rho_d (1-x)^{d-1} \\
&\geq \sum_{d=\mathtt{r}_{\min}}^{\mathtt{r}_{\max}} (1-(d-1)x)\rho_d \\
&= 1 - (\mathtt{r}_{\mathrm{avg}} - 1)x \\
\Rightarrow 1 - \rho(1-x) &\leq (\mathtt{r}_{\mathrm{avg}} - 1)x
\end{aligned}$$

where $\mathtt{r}_{\mathrm{avg}} = \sum_j j\rho_j$ is the average check-node degree from the edge-perspective. For

$0 \le (\mathtt{r}_{\text{avg}} - 1)x \le 1,$

$$\begin{aligned} f(\varepsilon, x) &= \varepsilon\lambda(1 - \rho(1 - x)) \\ &\overset{\text{a}}{\le} \varepsilon\lambda((\mathtt{r}_{\text{avg}} - 1)x) \\ &= \varepsilon \sum_{i=\mathtt{l}_{\min}}^{\mathtt{l}_{\max}} \lambda_i((\mathtt{r}_{\text{avg}} - 1)x)^{i-1} \\ &\overset{\text{b}}{\le} \varepsilon \sum_{i=\mathtt{l}_{\min}}^{\mathtt{l}_{\max}} \lambda_i((\mathtt{r}_{\text{avg}} - 1)x)^{\mathtt{l}_{\min}-1} \\ \Rightarrow f(\varepsilon, x) &\le \varepsilon((\mathtt{r}_{\text{avg}} - 1)x)^{\mathtt{l}_{\min}-1} =: g(\varepsilon, x) \end{aligned} \tag{20}$$

Note that (a) follows from the monotonicity of $\lambda(x)$, and (b) follows from the given condition $0 \le (\mathtt{r}_{\text{avg}} - 1)x \le 1$. To make the notation easier, let us denote $A = \varepsilon(\mathtt{r}_{\text{avg}} - 1)^{\mathtt{l}_{\min}-1}$. Since we are operating in the region $\varepsilon < \varepsilon_{\text{th}}$ where x_t converges to zero, there exists an R such that $Ax_R^{\mathtt{l}_{\min}-2} \le 1$ and $(\mathtt{r}_{\text{avg}} - 1)x_R \le 1$. The first inequality will be used later in the proof.

Let us construct a sequence $z_{R+i+1} = g(\varepsilon, z_{R+i})$ with $z_R = x_R$. It is immaterial what z_i takes when $i < R$. We then claim that $x_{R+i} \le z_{R+i}$ for any non-negative integer i. We can prove this claim by induction. The base case is when $i = 0$ and it is true by our choice of z_R. Assuming the claim is true for some integer $i \ge 0$, we have

$$x_{R+i+1} = f(\varepsilon, x_{R+i}) \le g(\varepsilon, x_{R+i}) \le g(\varepsilon, z_{R+i}) = z_{R+i+1}$$

The first inequality is due to (20) and the second inequality is due to the monotonicity of g and the induction hypothesis. This proves the claim.

We now have,

$$
\begin{aligned}
z_{R+1} &= A z_R^{\mathtt{l}_{\min}-1} \\
z_{R+i} &= A^{1+(\mathtt{l}_{\min}-1)+(\mathtt{l}_{\min}-1)^2+\cdots+(\mathtt{l}_{\min}-1)^{i-1}} z_R^{(\mathtt{l}_{\min}-1)^i} \\
&= A^{\frac{(\mathtt{l}_{\min}-1)^i-1}{\mathtt{l}_{\min}-2}} z_R^{(\mathtt{l}_{\min}-1)^i} \\
&= A^{\frac{-1}{\mathtt{l}_{\min}-2}} \left(A^{\frac{1}{\mathtt{l}_{\min}-2}} x_R\right)^{(\mathtt{l}_{\min}-1)^i} \\
&= A^{\frac{-1}{\mathtt{l}_{\min}-2}} \exp\left((\mathtt{l}_{\min}-1)^i \left(\frac{\log A}{\mathtt{l}_{\min}-2} + \log x_R\right)\right) \\
&= A^{\frac{-1}{\mathtt{l}_{\min}-2}} \exp\left(-\alpha_R (\mathtt{l}_{\min}-1)^i\right)
\end{aligned}
$$

Due to our choice of R, $\alpha_R \triangleq \frac{-1}{\mathtt{l}_{\min}-2}\log A - \log x_R$ is positive. For $t \geq R$, we have

$$
\begin{aligned}
x_t &\leq z_t \\
&= A^{\frac{-1}{\mathtt{l}_{\min}-2}} \exp\left(-\alpha_R (\mathtt{l}_{\min}-1)^{t-R}\right) \\
&= A^{\frac{-1}{\mathtt{l}_{\min}-2}} \exp\left(-\frac{\alpha_R}{(\mathtt{l}_{\min}-1)^R}(\mathtt{l}_{\min}-1)^t\right) \\
&= A^{\frac{-1}{\mathtt{l}_{\min}-2}} \exp\left(-\beta(\mathtt{l}_{\min}-1)^t\right)
\end{aligned}
$$

Note that

$$
\beta \triangleq \frac{\alpha_R}{(\mathtt{l}_{\min}-1)^R} > 0
$$

Therefore, we have

$$
x_t = \mathcal{O}\left(\exp(-\beta(\mathtt{l}_{\min}-1)^t)\right) \quad \text{as } t \to \infty
$$

To prove the second half, we note that for $x \in [0,1]$

$$
\begin{aligned}
L(x) &= \sum L_i x^i \leq \sum L_i x^{i-1} \\
&= \frac{1}{\int_0^1 \lambda(x)\mathrm{d}x} \sum_{i=\mathtt{l}_{\min}}^{\mathtt{l}_{\max}} \frac{\lambda_i}{i} x^{i-1} \\
&\leq \frac{1}{\mathtt{l}_{\min}\int_0^1 \lambda(x)\mathrm{d}x}\lambda(x) \\
\Rightarrow y_t &\leq \frac{1}{\mathtt{l}_{\min}\int_0^1 \lambda(x)\mathrm{d}x} x_t \\
\Rightarrow y_t &= \mathcal{O}\left(\exp(-\beta(\mathtt{l}_{\min}-1)^t)\right)
\end{aligned}
$$

□

It is important to note that a similar *double exponential* decay result is not true for DDPs that have degree-two variable nodes, i.e., with $\mathtt{l}_{\min} = 2$. Working out the expressions for this case, we can see that x_t and y_t exhibit only an *exponential* decay as the number of iterations t increases. Note that x_t is the expectation of the root-node bit-error probability taken over the possible outcomes of the tree ensemble $\vec{\mathcal{T}}_t$. The dominating term in this expectation is the contribution of the worst-case trees, namely, the trees that contain a long chain of nodes such that the only participating variable nodes are the ones with the minimum left degree $\mathtt{l}_{\min}$. DDPs with $\mathtt{l}_{\min} = 2$ form a special case where the contribution by the worst-case trees decays only exponentially fast in t.

5.3 Motivation for Large-Girth LDPC Codes

Suppose we are given a DDP (λ, ρ) with $\mathtt{l}_{\min} \geq 3$. For $k = 1, 2, 3, \ldots$, let (n_k) be a strictly increasing sequence of positive integers and let t_k be such that

$$t_k = \left\lceil \frac{\log\log n_k + \log a - \log \beta}{\log(\mathtt{l}_{\min} - 1)} \right\rceil$$

for any positive integer a. By Lemma 5.2, we have $y_{t_k} = \mathcal{O}\left(1/n_k^a\right)$. In particular, we have $y_t = \mathcal{O}\left(1/n^3\right)$ for $a = 3$ (we drop the subscript k for convenience). Since the density evolution estimate y_t is only an approximation of the actual BER of the standard ensemble, this does not necessarily mean that the actual bit-error probability $y(t, n)$ itself decays as $\mathcal{O}(1/n^3)$. In particular, note that we want to analyze the BER under BP decoding as the number of iterations t increases with n.

There are only a few rigorous results regarding the "closeness" of the density evolution approximation and to our knowledge, none of these fit our requirement. For example, we know the following results

- For $\mathcal{G}(n, \lambda, \rho)$

$$\lim_{n\to\infty} x(t, n) = x_t, \qquad \lim_{n\to\infty} y(t, n) = y_t$$

as long as t remains constant [34, Thm. 3.49]. We cannot use this result directly since we let t grow with n.

- Korada and Urbanke [45] consider a randomly selected code $\mathcal{C}_n$ from $\mathcal{G}(n, \lambda, \rho)$ and analyze the expected BER as the iteration count $t(n)$ increases monotonically with the block length n. In particular, they show that

$$\lim_{n\to\infty} \mathbb{E}\left(P_b^{\mathrm{BP}}\left(\mathcal{C}_n, \varepsilon, t(n)\right)\right) = 0$$

whenever either of the following conditions are met.

1. The minimum variable-node degree is at least five and the channel parameter is below the threshold, i.e.,

$$\mathtt{l}_{\min} \geq 5 \qquad\qquad \varepsilon < \varepsilon_{\mathrm{th}}$$

2. The minimum variable-node degree is at least three and the channel parameter is below a quantity $\bar{\varepsilon}_{\mathrm{th}}$, which is less than the threshold $\varepsilon_{\mathrm{th}}$, i.e.,

$$\mathtt{l}_{\min} \geq 3 \qquad\qquad \varepsilon < \bar{\varepsilon}_{\mathrm{th}}$$

The above result states the regions where $y(t, n)$ converges to zero when t is a non-decreasing function of n. However, for our strong secrecy result, we must also know that speed at which it converges to zero.

To achieve strong secrecy, we must find some ensemble $\mathcal{H}_n$ for which $y(t, n) = \mathcal{O}(1/n^3)$, where t is growing with n at least as fast as $\log\log n$. In general, this is not true for $\mathcal{G}(n, \lambda, \rho)$. For example, any irregular DDP with $\mathtt{l}_{\min} = 3$ does not satisfy $y(t, n) = \mathcal{O}(y_t)$ any $\varepsilon > 0$. In particular, from one of the results by Orlitsky, et al. [39, Thm. 16] we can infer that the following for the standard ensemble

$$y(t, n) \geq \lim_{t\to\infty} y(t, n) = \mathbb{E}P_B^{\mathrm{BP}}(\mathcal{C}_n, \varepsilon) = \Theta\left(\frac{1}{n^{\lceil \frac{\mathtt{l}_{\min}}{2} \rceil - 1}}\right)$$

for $\mathtt{l}_{\min} \geq 2$ and $\varepsilon < \varepsilon_{\text{th}}$. This means that we require the minimum variable-node degree $\mathtt{l}_{\min}$ to be at least five for strong secrecy. However, this reduces the erasure threshold ε_{th} to low values. From the discussion in Chapter 3, it is clear that we must have very high values of ε_{th} to achieve strong secrecy at high rates. Our objective is to achieve fast block-error (bit-error) decay and high thresholds simultaneously, and the standard ensemble of LDPC codes is not adequate for these requirements. Therefore, we must construct special ensembles of LDPC codes.

5.3.1 Strong Secrecy Using Large-Girth Regular LDPC Codes

Let $\mathcal{G}_g(n, \lambda, \rho)$ denote the subset of Tanner graphs in $\mathcal{G}(n, \lambda, \rho)$ whose girth is more than g. Clearly, the level-t computation graphs of $\mathcal{G}_{4t}(n, \lambda, \rho)$ are cycle free. This means that any possible outcome of $\mathring{\mathcal{C}}_t$ is also a possible outcome of $\mathring{\mathcal{T}}_t$; in other words, the random trees $\mathring{\mathcal{C}}_t$ and $\mathring{\mathcal{T}}_t$ have the same "support." This does not necessarily mean that $\mathring{\mathcal{C}}_t$ and $\mathring{\mathcal{T}}_t$ are identically distributed and therefore, $y(t, n) = y_t$ is not necessarily true for $\mathcal{G}_{4t}(n, \lambda, \rho)$.

The regular LDPC code ensemble $\mathcal{G}_{4t}(n, x^{c-1}, x^{d-1})$ is a special case for which $\mathring{\mathcal{C}}_t$ and $\mathring{\mathcal{T}}_t$ are equal to a unique tree T. This is because the girth condition forces the two random variables to have the same support and the regularity of the code forces that support to be of size one. Since $y(t, n)$ is calculated from $\mathring{\mathcal{C}}_t$ in the same way as y_t is calculated from $\mathring{\mathcal{T}}_t$, we have $y(t, n) = y_t$. Using a similar reasoning, we can also say that $x(t, n) = x_t$.

In essence, density evolution analysis is approximate because it makes the following assumptions.

1. The decoding neighborhood is a tree.

2. The node degrees in the decoding tree are independent.

For large-girth Tanner graphs, the first assumption is true. For large-girth *regular*

Table 1: The validity of density evolution assumptions for different ensembles of LDPC codes.

Ensemble	$\mathring{\mathcal{C}}_t$ is a tree?	$\mathring{\mathcal{C}}_t = \mathring{\mathcal{T}}_t$ statistically?
Standard enesemble	✗	✗
Large-girth irregular	✓	✗
Large-girth regular	✓	✓

Tanner graphs, there is a unique decoding neighborhood with only one choice for variable-node degrees and only one choice for check-node degrees, and hence the second assumption is also true. However, the second assumption is not justified for large-girth irregular Tanner graphs. Therefore, we are able to assert that the density evolution estimate is exact in the case of large-girth regular LDPC codes, but are unable to do the same for the irregular counterpart. This means that large-girth irregular LDPC codes require a much closer analysis to prove any strong secrecy results.

Assume that there exists a sequence $(\mathcal{C}_n^{\perp})$ of (c,d)-regular LDPC codes with $c \geq 3$ such that their Tanner graphs have girth more than $4t$, where

$$t = \left\lceil \frac{\log\log n + \log 3 - \log \beta}{\log(c-1)} \right\rceil$$

(The existence of such codes will be proved in the next section). For these codes, we have

$$P_b^{\mathrm{BP}}(\mathcal{C}_n^{\perp}, \varepsilon, t) = y(t,n) = y_t = \mathcal{O}\left(\frac{1}{n^3}\right)$$

for $\varepsilon < \varepsilon_{\mathrm{th}}$. Here, $P_b^{\mathrm{BP}}(\mathcal{C}_n^{\perp}, \varepsilon, t)$ denotes the bit-error probability after t iterations. By the above equation, the coset coding scheme using the dual sequence $(\mathcal{C}_n)$ will achieve strong secrecy on BEWC(ξ) for $\xi > 1 - \varepsilon_{\mathrm{th}}$.

5.3.2 Existing Constructions for LDPC Codes with High Girth

Construction of LDPC codes with high girth (not necessarily with logarithmic growth) has received significant attention from the coding theory community. We know of the

following constructions.

1. The progressive edge growth (PEG) algorithm by Hu, et al. [46] constructs LDPC codes with a prescribed left (variable-node) degree distribution and rate. The PEG algorithm, which is a greedy algorithm, starts with a Tanner graph with no edges, and for each variable node, it chooses the farthest away check node and adds an edge to it. If two check nodes are equidistant from the given variable node, then the one with the least degree is chosen. In this way, edges are added to satisfy all the left degree requirements.

   The LDPC codes produced by PEG have a very good error-correcting performance over additive white Gaussian noise (AWGN) channels, and empirical evidence shows that PEG can create Tanner graphs with high girth even for short block lengths. Under the assumption that the maximum right degree is bounded, Hu, et al. [46] proved that PEG can create large-girth Tanner graphs. However, we are unable to ascertain whether this assumption about the right degrees is true. Therefore, we are unable to conclude that PEG is a large-girth construction.

2. An algorithm to construct near-regular graphs with large-girth was proposed by Chandran [47]. The ideas behind this algorithm are very similar to PEG. For a given positive integer k and an arbitrarily large integer n, this algorithm constructs a graph in n vertices with minimum degree $k-1$, maximum degree $k+1$ and average degree k such that the girth of the graph is lower-bounded by $\log_k(n) + \mathcal{O}(1)$ as n increases.

   The above algorithm was modified by Krishnan, et al. [48] to construct large-girth near-regular LDPC codes. Given the integers c, d with $1 < c < d$, and increasing block length n, the modified algorithm constructs large-girth Tanner graphs with variable-node degrees in the set $\{c-1, c, c+1\}$ with average c, and

check-node degrees in the set $\{d-1, d, d+1\}$ with average d.

3. For any odd number $k \geq 3$ and a prime power q, Lazebnik and Ustimenko [49] gave an algebraic construction of q-regular bipartite graphs on $2q^k$ vertices with girth at least $k+5$. It was later shown in [50] that this graph is disconnected and its connected components are isomorphic. For $k \geq 6$, it was shown that each component has $2q^{k-\lfloor \frac{k+2}{4} \rfloor+1}$ vertices. The graphs produced by this algorithm are large-girth graphs. Based on these graphs and their underlying algebraic structure, Kim, et al. [51] constructed LDPC codes having asymptotically large girth. It can be noted that the existing code constructions using this graph produce codes that either have rate $1/q$ or have rate very close to 1.

4. Margulis [52] constructed $2r$-regular graphs of girth at least $c(\log n)/(\log r)$ on n vertices for some constant $c > 0$. In the same work, he also outlined a method to construct rate-$\frac{1}{2}$ regular LDPC codes using these graphs.

5. Based on Margulis' idea, Rosenthal and Vontobel [53, 54] constructed large-girth regular LDPC codes using the large-girth graphs by Lubotzky, Phillips and Sarnak [55].

6. A construction of high girth regular LDPC codes was proposed by Gallager [38, Appendix C] in his monograph. In his construction, a parity-check matrix is built heuristically to avoid short cycles.

It can be noted that the above constructions produce only LDPC codes of specific rates and specific (regular) degree distributions. For our problem, we require an ensemble of large-girth LDPC codes with arbitrary irregular degree distributions and to our knowledge, none of the prior constructions fit our requirement.

5.4 LPS Graphs—Background

In [55], Lubotzky, Phillips and Sarnak published a construction of a family of regular graphs that have the large-girth property. In this section, we provide a very brief overview of these graphs; for a more detailed discussion, we point the reader to the book by Davidoff, et al. [56]. We shall call these graphs *LPS graphs*, after the inventors. LPS graphs belong to the class *Cayley graphs*. Given a group G and an inverse-closed subset S of G, i.e., $s^{-1} \in S,\ \forall s \in S$, the Cayley graph $\Gamma(G, S)$ is an undirected simple graph defined as follows:

- The vertex set of $\Gamma(G, S)$ is G.
- For any $g \in G$ and $s \in S$, there is an edge between g and gs.

5.4.1 Construction

The LPS graphs $X^{p,q}$ are defined for odd primes p and q with $q > 2\sqrt{p}$. To construct $X^{p,q}$, one first chooses a set $S_{p,q}$ with $p+1$ elements from the projective linear group $\mathrm{PGL}_2(q)$ of invertible 2×2 invertible matrices over $\mathbb{F}_q$. We will not discuss how $S_{p,q}$ is chosen, since it is beyond the scope of this dissertation. The LPS graph $X^{p,q}$ is constructed as follows.

1. If p is a quadratic residue modulo q, then $X^{p,q} = \Gamma\left(\mathrm{PSL}_2(q), S_{p,q}\right)$.
2. If p is a quadratic non-residue modulo q, then $X^{p,q} = \Gamma\left(\mathrm{PGL}_2(q), S_{p,q}\right)$.

Here $\mathrm{PSL}_2(q)$ is the special linear group of 2×2 invertible matrices over $\mathbb{F}_q$.

5.4.2 Properties

LPS graphs belong to the class of *Ramanujan graphs*, defined subsequently. The adjacency matrix of a simple graph with n vertices is an $n \times n$ matrix $[a_{i,j}]$ such that $a_{i,j} = 1$ whenever vertices i and j are adjacent, and $a_{i,j} = 0$ otherwise. Consider the

eigenvalues of the adjacency matrix. For a k-regular graph, any eigenvalue μ is such that $|\mu| \leq k$. A Ramanujan graph is a k-regular graph such that if μ is an eigenvalue and $|\mu| \neq k$, then $|\mu| \leq 2\sqrt{k-1}$.

LPS graphs also have the large-girth property, as noted by the following.

Theorem 5.3 ([56, Thm. 4.2.2]). *Let p, q be distinct, odd primes, with $q > 2\sqrt{p}$. The graphs $X^{p,q}$ are $(p+1)$-regular graphs that are connected and Ramanujan. Moreover,*

1. *If p is a quadratic residue modulo q, then $X^{p,q}$ is a non-bipartite graph with $\frac{q(q^2-1)}{2}$ vertices, satisfying the girth estimate* $\text{girth}(X^{p,q}) \geq 2\log_p q$

2. *If p is a quadratic non-residue modulo q, then $X^{p,q}$ is a bipartite graph with $q(q^2-1)$ vertices, satisfying* $\text{girth}(X^{p,q}) \geq 4\log_p q - \log_p 4$

The large-girth property of LPS graphs is independent of their Ramanujan property. In fact, Ramanujan graphs containing loops exist [57].

5.4.3 Applications in Error-Correction Coding

LPS graphs have good expansion, which is a direct consequence of their Ramanujan property. The expansion property of these graphs was used by Sipser and Spielman [58] to construct asymptotically good LDPC codes that are also expanders. The large-girth property of LPS graphs was used to construct large-girth LDPC codes by Rosenthal and Vontobel [54].

5.5 Construction of Large-Girth Tanner Graphs

In this section, we will describe our construction of large-girth Tanner graphs for an arbitrary DDP (λ, ρ). Our construction works for both regular and irregular DDPs and it relies on existing constructions of large-girth regular graphs, which may or may not be bipartite.

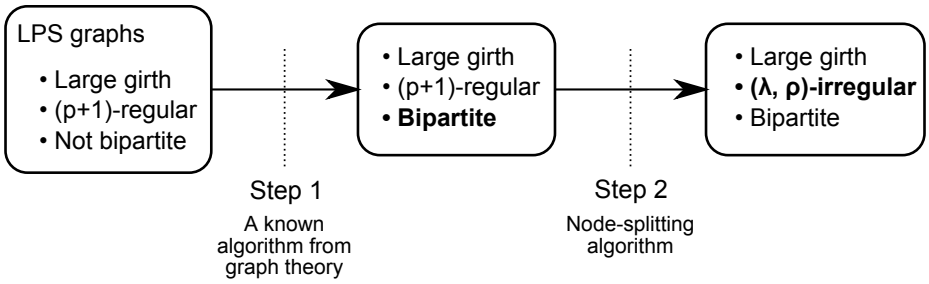


Figure 24: An overview of the algorithm to construct large-girth LDPC codes.

The following is a high-level description of our algorithm (illustrated in Fig. 24). Given a DDP (λ, ρ), we start with a sequence of large-girth $(p+1)$-regular graphs for a suitable integer p. In the first step, we use a well-known algorithm from the graph theory community to convert these into bipartite graphs; we create a sequence of large-girth $(p+1)$-regular bipartite graphs from the given sequence of graphs. In the second step, we use a *node-splitting* algorithm to create nodes of smaller degrees in such a way that the resulting graph is a (λ, ρ) Tanner graph. For a given node of degree $(p+1)$, we create new nodes and reassign some of the edges from the given node to the newly created nodes. Note that in both the steps, we increase the number of nodes linearly without decreasing the girth. This maintains the logarithmic growth in the girth, thereby giving rise to a sequence of large-girth (λ, ρ) LDPC codes. Our algorithm can start with *any* sequence of large-girth regular graphs and the results in this chapter are true regardless of our starting sequence. For the sake of clarity, we will only discuss the case where we start with LPS graphs.

5.5.1 Large-Girth Bipartite Graphs from Large-Girth Graphs

The first step in our algorithm is to create a large-girth $(p+1)$-regular bipartite graph from the LPS graph $X^{p,q}$. For this purpose, we make use of an algorithm from the book *Extremal Graph Theory* by Bollobás [59]. This algorithm is outlined as Algorithm 5.1 in this dissertation. Given a simple graph G, this algorithm creates a

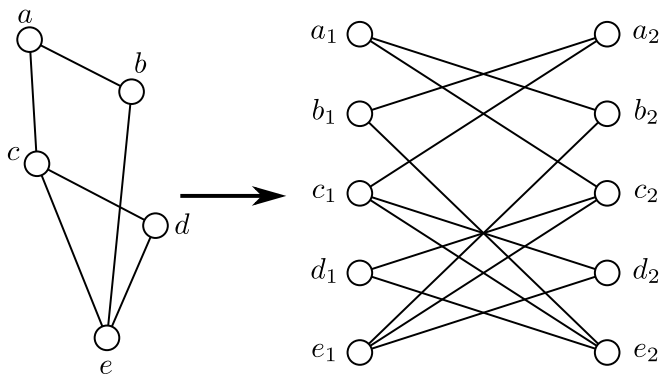


Figure 25: An illustration of Algorithm 5.1 to create bipartite graphs.

bipartite graph $B(G)$ with twice the number of vertices and edges.

Algorithm 5.1 Constructing a bipartite graph given any graph [59, §3.1].

1: Given a graph G in n vertices, create an identical copy G' with $V(G) \cap V(G') = \varnothing$. Let $f : V(G) \to V(G')$ be a graph homomorphism.
2: Create a graph H with vertex set $V(H) = V(G) \cup V(G')$ and edge set $E(H) = \{\{x, y\} : x \in V(G), y \in V(G'), f(x) \sim y \text{ in } G'\}$. That is, if $a_1, b_1 \in V(G)$, $a_2 = f(a_1)$, $b_2 = f(b_1)$ and $a_1b_1 \in E(G)$ (or equivalently, if $a_2b_2 \in E(G')$), then $a_1b_2, a_2b_1 \in E(H)$.

Lemma 5.4. *Given a graph G, if $H = B(G)$ then* $\text{girth}(H) \geq \text{girth}(G)$.

Proof. For any cycle C in H with the vertices in the order

$$(u_0, v_0, u_1, v_1, \ldots, u_{r-1}, v_{r-1}, u_0)$$

there exists a closed walk

$$W = (u_0, f^{-1}(v_0), u_1, f^{-1}(v_1), \ldots, u_{r-1}, f^{-1}(v_{r-1}), u_0)$$

in G. Note that $r \geq 2$. We show that this closed walk W contains a cycle. We do this by showing that while traversing W, we do not retrace an edge, i.e., we do not encounter an edge twice in succession.

Suppose otherwise. Let v_i be the vertex where we traceback. The sequence $u_i, v_i, u_{(i+1) \bmod r}$ is such that $u_i = u_{(i+1) \bmod r}$. This is a contradiction since all the vertices in the original cycle are distinct and $r \geq 2$.

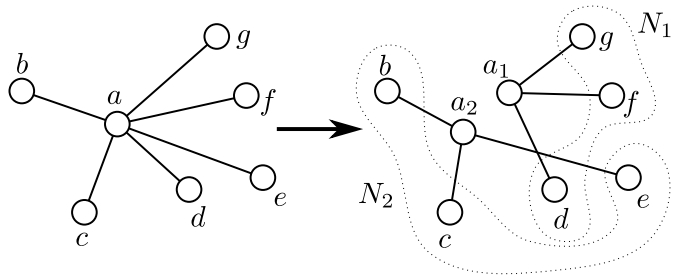


Figure 26: An illustration of Algorithm 5.2 to split a vertex.

Therefore, W contains a cycle C^*. We have

$$\text{length}(C) \geq \text{length}(C^*) \geq \text{girth}(G)$$

which proves that $\text{girth}(H) \geq \text{girth}(G)$. Note that the second inequality in the above equation follows from the fact that C^* is a cycle in G. □

5.5.2 Large-Girth Tanner Graphs from Large-Girth Regular Bipartite Graphs

The Basic Node-Splitting Step

The basic step involved in creating vertices of smaller degrees from a given vertex of larger degree is the node-splitting step, Algorithm 5.2. This step is illustrated in Fig. 26, where a node of degree six is split into two nodes of degrees three each.

Algorithm 5.2 Splitting a vertex into vertices of smaller degrees.

1: Given a vertex v in a graph G, we partition the set of all its neighbors $N(v)$ into $N_1, N_2, \ldots, N_s$.
2: We create a new graph H by deleting v from G and adding new vertices $v_1, v_2, \ldots, v_s$ and connecting v_i to the vertices in N_i for all i.

In this work, we will only consider the creation of equal sized partitions N_i in Step 1 of Algorithm 5.2. Under this restriction, the partitioning of $N(v)$ can be done in two ways.

- **Deterministic version.** Let $(e_1, e_2, \ldots, e_M)$ be a simple ordering of the edges in G. If $N(v) = \{e_{i_1}, e_{i_2}, \ldots, e_{i_{sk}}\}$ with $i_1 \leq i_2 \leq \cdots \leq i_{sk}$, we choose $N_1 = \{e_{i_1}, e_{i_2}, \ldots, e_{i_k}\}$, $N_2 = \{e_{i_{k+1}}, e_{i_{k+2}}, \ldots, e_{i_{2k}}\}$, etc.

- **Random version.** The partitioning of $N(v)$ is done in a random fashion.

Though the girth properties of the graphs obtained by both the versions of Algorithm 5.2 are similar, it is easier to count graphs of a particular configuration when we use the deterministic version than when we use the random version. The deterministic version will be used in our construction of irregular Tanner graphs. In the following lemma, we show that node-splitting does not decrease girth.

Lemma 5.5. *Given a graph G, if H is a graph obtained by splitting an arbitrary vertex of G according to either version of Algorithm 5.2, then* $\text{girth}(H) \geq \text{girth}(G)$.

Proof. Suppose H does not have any cycles. In this case, $\text{girth}(H) = \infty$ and the lemma is true.

We are now left with the case where H has cycles. Consider any cycle C in H. Let v be the vertex of G that is being split and let $V_{\text{new}} = \{v_1, v_2, \ldots, v_s\}$ be the set of new vertices created. By traversing along C and identifying vertices v_i with v, we will get a closed walk W in G. We show that W contains a cycle.

If C has less than two vertices from the set V_{new}, then W is a cycle and we are done. Otherwise, C has at least two of these vertices. We can pick v_i, v_j such that while traversing from one to the other along C, we don't encounter any other vertices from V_{new}. Let this path (excluding v_i, v_j) be denoted by P. Since v_i and v_j are not adjacent and $N(v_i) \cap N(v_j) = \varnothing$, this path has at least two vertices. Therefore, vPv is a cycle C^* in W that is smaller than C. Since G contains the cycle C^*, we have $\text{length}(C) \geq \text{length}(C^*) \geq \text{girth}(G)$, which shows that $\text{girth}(H) \geq \text{girth}(G)$. □

Note that Algorithm 5.2 can sometimes create a disconnected graph. That is, H may be disconnected even if G is connected. However, we can see that Lemma 5.5 is valid regardless of any disconnections introduced by node splitting. Furthermore, the proof of the main result in this chapter relies only on Lemma 5.5 and is valid even though some of the Tanner graphs in our ensemble may be disconnected.

Regular Bipartite Graphs with Arbitrary Degree

When p is a quadratic residue modulo q, we can use the construction in Algorithm 5.1 to generate a bipartite $(p+1)$-regular graph in $q(q^2-1)$ vertices with girth at least $2\log_p q$. Therefore, given primes p and q with $q > 2\sqrt{p}$, it is possible to construct a $(p+1)$-regular bipartite graph in $q(q^2-1)$ vertices with girth at least $2\log_p q$.

In our overall construction, we want to split the vertices of a k-regular graph into vertices of smaller degree. It is easier to work with large-girth k-regular bipartite graphs for arbitrary k, than it is with large-girth $(p+1)$-regular bipartite graphs for prime p. With this in mind, we propose Algorithm 5.3 to create the former from the latter using node-splitting. In this algorithm, we first find an integer s such that there exists a prime p with $p+1 = sk$. Then, we split each vertex of the $(p+1)$-regular bipartite graph into s new vertices each to get a k-regular bipartite graph. The existence of s (and p) is guaranteed by a corollary to the following.

Theorem 5.6 (Dirichlet's Theorem on Arithmetic Progressions [60, Ch. 7])**.** *Given two positive integers a, b that are relatively prime, i.e.,* $\gcd\{a,b\} = 1$*, the sequence $(an+b)_{n\in\mathbb{N}}$ contains an infinite number of primes.*

By observing that $\gcd\{k, k-1\} = 1$, we have the following.

Corollary 5.7. *Given any positive integer k, there are infinite number of primes of the form $sk-1$, with $s \in \mathbb{N}$.*

Large-Girth Regular Tanner Graphs

Recall from §5.3.1 that for the case of large-girth regular LDPC codes, the security result is straightforward. We provide Algorithm 5.4 to construct large-girth regular Tanner graphs for a given left degree c and right degree d.

Algorithm 5.3 Constructing large-girth k-regular bipartite graphs.

1: Given a positive integer k, find the smallest solution for $s \in \mathbb{N}$ such that $p = sk - 1$ is a prime. The existence of s is guaranteed by Dirichlet's Theorem on Arithmetic Progressions. Denote $sk - 1$ by p.
2: Pick a sequence of primes greater than $2\sqrt{p}$. For each such prime q, generate the LPS graph $X^{p,q}$.
3: If p is a quadratic residue modulo q, then $G = B(X^{p,q})$. Otherwise, $G = X^{p,q}$. In either case, G is an (sk)-regular bipartite graph on $q(q^2 - 1)$ vertices and $\text{girth}(G) \geq 2\log_p q$.
4: Split each vertex of G successively into s vertices of degree k according to either version of Algorithm 5.2. The resulting graph H is a k-regular bipartite graph with $sq(q^2 - 1)$ vertices and $\text{girth}(H) \geq \text{girth}(G) \geq 2\log_p q$.

Algorithm 5.4 Constructing large-girth (c, d)-regular bipartite graphs.

1: Let $k = \text{LCM}\{c, d\}$. Construct a sequence of k-regular bipartite graphs of large girth according to Algorithm 5.3.
2: Given a k-regular bipartite graph G with $sq(q^2-1)$ vertices and $\text{girth}(G) \geq 2\log_p q$, let (V_v, V_c) be the bipartition of the vertices. We have $|V_v| = |V_c| = \frac{sq(q^2-1)}{2}$.
3: Split each vertex in V_v into k/c new vertices of degree c each according to either version of Algorithm 5.2 to get $\frac{k}{c}\frac{sq(q^2-1)}{2}$ left vertices of degree c.
4: Split each vertex in V_c into k/d new vertices of degree d each according to either version of Algorithm 5.2 to get $\frac{k}{d}\frac{sq(q^2-1)}{2}$ right vertices of degree d. The resultant graph H is a (c, d)-regular bipartite graph with $\text{girth}(H) \geq 2\log_p q$.

Large-Girth Irregular Tanner Graphs

For an *arbitrary* sequence of large-girth irregular LDPC codes, we are unable to show that the girth property directly implies a strong secrecy result. However, we are able to prove the same for large-girth irregular LDPC codes constructed in a specific manner, namely by Algorithm 5.5. In Algorithm 5.5, we create a large-girth k-regular bipartite graph G according to Algorithm 5.3. Then, we randomly order the nodes and split them into nodes of smaller degrees such that contiguous blocks give rise to daughter nodes of identical degree.

5.6 Asymptotic BER of Large-Girth LDPC Codes

For a given DDP (λ, ρ), we can create a sequence of large-girth (λ, ρ)-irregular LDPC codes $(\mathcal{C}_n)$ of increasing block-length n using Algorithm 5.5. We denote the large-girth graphs associated with $\mathcal{C}_n$ by $\mathcal{R}_n$.

Theorem 5.8. *For a given DDP (λ, ρ) with minimum left degree $\mathtt{l}_{\min} \geq 3$, the sequence of large-girth (λ, ρ)-irregular LDPC codes $(\mathcal{C}_n)$ created using Algorithm 5.5 is such that whenever $\varepsilon < \varepsilon_{\text{th}}$ we have*

$$\mathbb{E}P_b^{\text{MP}}(\mathcal{C}_n, \varepsilon) = \mathcal{O}\left(\exp(-c_1 n^{c_2})\right) \tag{21}$$

for some positive constants c_1, c_2.

5.6.1 Proof of Theorem 5.8

The only sources of randomness in Algorithm 5.5 for a given large-girth graph G (at the end of Step 3) are the permutation functions σ and π. The probability distribution of $\mathcal{R}_n$ given G is easier to analyze than that of $\mathcal{R}_n$ when G is not specified. Clearly, (21) is true whenever

$$\mathbb{E}\left(P_b^{\text{MP}}(\mathcal{C}_n, \varepsilon)|G\right) = \mathcal{O}\left(\exp(-c_1 n^{c_2})\right) \tag{22}$$

Algorithm 5.5 Constructing large-girth (λ, ρ) irregular bipartite graphs.

1: Let k be the least common multiple (LCM) of all the left and right degrees. Let a be the smallest positive integer such that $a\lambda_i, a\rho_j \in \mathbb{N}$ for all i, j.

2: Let s be the smallest natural number such that $sak-1$ is a prime number. Call this prime number p. Choose an arbitrary prime $q > 2\sqrt{p}$. Construct an (ak)-regular bipartite graph G_0 according to Algorithm 5.3. The graph G_0 has $sq(q^2-1)$ vertices and $\text{girth}(G_0) \geq 2\log_p q$.

3: Split each vertex of G_0 into a vertices of degree k by successively applying Algorithm 5.2 (either version) and denote the resulting k-regular bipartite graph by G. The graph G has n_0 vertices on the left and n_0 vertices on the right, where $n_0 = \frac{asq(q^2-1)}{2}$, and $\text{girth}(G) \geq 2\log_p q$.

4: Let $(v_1, v_2, \ldots, v_{n_0})$ be some ordering of the "left" vertices in G and let $(c_1, c_2, \ldots, c_{n_0})$ be some ordering of the "right" vertices in G. Also, let $(e_1, e_2, \ldots, e_{n_0 k})$ be some ordering of the edges in G.

5: Let σ and π be two randomly chosen permutation functions over the set $\{1, 2, \ldots, n_0\}$.

6: Consider the ordered set $(v'_1, v'_2, \ldots, v'_{n_0})$, where $v'_i = v_{\sigma(i)}$. In this ordered set,

- split the first $n_0\lambda_{\mathtt{l}_{\min}}$ vertices into $n_0 k\lambda_{\mathtt{l}_{\min}}/\mathtt{l}_{\min}$ vertices of degree $\mathtt{l}_{\min}$,
- split the next $n_0\lambda_{\mathtt{l}_{\min}+1}$ vertices into $n_0 k\lambda_{\mathtt{l}_{\min}+1}/(\mathtt{l}_{\min}+1)$ vertices of degree $\mathtt{l}_{\min}+1$,
- ...
- split the last $n_0\lambda_{\mathtt{l}_{\max}}$ vertices into $n_0 k\lambda_{\mathtt{l}_{\max}}/\mathtt{l}_{\max}$ vertices of degree $\mathtt{l}_{\max}$.

In the above, we split the vertices according to the deterministic version of Algorithm 5.2.

7: Do a similar operation for the check nodes using the ordered set $(c'_1, c'_2, \ldots, c'_{n_0})$, where $c'_j = c_{\pi(j)}$, and the distribution ρ. The resulting graph H is a (λ, ρ) irregular bipartite graph with

$$n = \frac{aksq(q^2-1)}{2} \int_0^1 \lambda \mathrm{d}x$$

vertices and girth at least $2\log_p q$.

is true *uniformly* for all possible G in Step 3 of Algorithm 5.5.

Note that $P_b^{\mathrm{MP}}(\mathcal{C}_n, \varepsilon)$ denotes the probability of bit-error after infinite iterations of the MP algorithm (or equivalently, when a stopping set is encountered). This probability is clearly less than the probability of bit-error after a finite number of iterations. Therefore (22) is true whenever

$$\mathbb{E}\left(P_b^{\mathrm{MP}}(\mathcal{C}_n, \varepsilon, t(n))|G\right) \leq A(n) = \mathcal{O}\left(\exp(-c_1 n^{c_2})\right) \tag{23}$$

is true for some function $t(n)$. The role played by the quantity $A(n)$ is to ensure that we are able to upper bound the left hand side uniformly in G. We pick $t(n) = a \log n$, where $a > 0$ is such that $\mathrm{girth}(\mathcal{R}_n) \geq 4a \log n + 2$. We know that a exists because of the large-girth property of $\mathcal{R}_n$. Let $a_{\max}$ be the maximum possible value for a.

Lemma 5.9. *For any $\delta \in (0,1)$, there exists a natural number N, dependent only on n, λ, ρ and δ, such that for all $n \geq N$ we have*

$$\mathbb{E}\left(P_b^{\mathrm{MP}}(\mathcal{C}_n, \varepsilon, t(n))|G\right) \leq \frac{1}{1-\delta} y_{t(n)}(\varepsilon) \tag{24}$$

where $y_{t(n)}(\varepsilon)$ is the quantity defined in Lemma 5.2.

We know from Lemma 5.2 that

$$\begin{aligned} y_{t(n)}(\varepsilon) &= \mathcal{O}\left(\exp\left(-\beta(\mathtt{1}_{\min} - 1)^{t(n)}\right)\right) \\ &= \mathcal{O}\left(\exp\left(-\beta n^{a \log(\mathtt{1}_{\min} - 1)}\right)\right) \end{aligned}$$

The above equation, along with Lemma 5.9, completes the proof of the theorem.

Proof of Lemma 5.9. Consider the computation graph $\mathring{\mathcal{C}}_t$ of $\mathcal{R}_n$ (we write t for $t(n)$ to simplify notation). Clearly, $\Pr(\mathring{\mathcal{C}}_t = T) > 0$ if and only if $\Pr(\mathring{\mathcal{T}}_t = T) > 0$.

Let T be any valid level-t tree in the sense that $\Pr(\mathring{\mathcal{T}}_t = T) > 0$. Let $P_e(T, \varepsilon)$ be the probability that the root node of T is in error when the tree code associated with

T is transmitted over BEC(ε) and decoded with t iterations of the MP decoder. Note the following two equations

$$y_t(\varepsilon) = \sum \Pr(\mathring{\mathcal{T}}_t = T) P_e(T, \varepsilon)$$
$$\mathbb{E}\left(P_b^{\mathrm{MP}}(\mathcal{C}_n, \varepsilon, t) | G\right) = \sum \Pr(\mathring{\mathcal{C}}_t = T | G) P_e(T, \varepsilon)$$

From the above, we can see that the proof is complete once we show that for all large enough n, we have

$$\Pr(\mathring{\mathcal{C}}_t = T | G) \leq \frac{1}{1-\delta} \Pr(\mathring{\mathcal{T}}_t = T)$$

Let T be a valid level-t tree with i_0 being the degree of the root node. Let this tree have p_i variable nodes of degree i (including the root node, but excluding the leaf nodes) and q_j check nodes of degree j. We have

$$\Pr(\mathring{\mathcal{T}}_t = T) = L_{i_0} \lambda_{i_0}^{p_{i_0}-1} \prod_{i=3, i\neq i_0}^{\mathtt{l}_{\max}} \lambda_i^{p_i} \prod_{j=2}^{\mathtt{r}_{\max}} \rho_j^{q_j} \tag{25}$$

Now, consider $\mathring{\mathcal{C}}_t$. The probability that the root node v has degree i_0 is clearly L_{i_0}. The i_0 edges incident with v in $\mathcal{R}_n$ will correspond to i_0 edges in G incident with u, the parent node of v. Let $b(1), b(2), \ldots, b(i_0)$ be the i_0 neighbors of u in G corresponding to those edges. Let $c(1), c(2), \ldots, c(i_0)$ be the daughter nodes in $\mathcal{R}_n$ corresponding to the same edges. The number of ways of choosing the permutation function π such that node $c(1)$ has degree j is equal to the number of ways of putting $b(1)$ into a slot that corresponds to degree j, which is $n_0 \rho_j$. Note that these slots are numbered. Here, $n_0 = 2n/\left(k \int_0^1 \lambda \mathrm{d}x\right)$ is the number of left (right) vertices in G, where k is the LCM of all the degrees in (λ, ρ).

In general, whenever T is a valid level-t tree, we have

$$\begin{aligned}\Pr(\mathring{\mathcal{C}}_t = T) = {} & L_{i_0} \tbinom{n_0\lambda_{i_0}-1}{p_{i_0}-1} (p_{i_0}-1)! \\ & \times \frac{\left(n_0 - 1 - \sum_{i=2}^{\mathtt{l}_{\max}} p_i\right)!}{n_0!} \prod_{i=\mathtt{l}_{\min}, i\neq i_0}^{\mathtt{l}_{\max}} \tbinom{n_0\lambda_i}{p_i} p_i! \\ & \times \frac{\left(n_0 - \sum_{j=\mathtt{r}_{\min}}^{\mathtt{r}_{\max}} q_j\right)!}{n_0!} \prod_{j=\mathtt{r}_{\min}}^{\mathtt{r}_{\max}} \tbinom{n_0\rho_j}{q_j} q_j! \end{aligned} \tag{26}$$

We note the following inequality.

$$\frac{\left(n_0 - 1 - \sum_{i=2}^{\mathtt{l}_{\max}} p_i\right)! \left(n_0 - \sum_{j=\mathtt{r}_{\min}}^{\mathtt{r}_{\max}} q_j\right)!}{n_0!\ n_0!} < \frac{1}{(n_0 - \sum p_i)^{(\sum p_i)-1} \quad (n_0 - \sum q_j)^{\sum q_j}} \tag{27}$$

We also see that

$$\begin{aligned} & L_{i_0} \tbinom{n_0\lambda_{i_0}-1}{p_{i_0}-1} (p_{i_0}-1)! \prod_{i=\mathtt{l}_{\min}, i\neq i_0}^{\mathtt{l}_{\max}} \tbinom{n_0\lambda_i}{p_i} p_i! \prod_{j=\mathtt{r}_{\min}}^{\mathtt{r}_{\max}} \tbinom{n_0\rho_j}{q_j} q_j! \\ & < L_{i_0} (n_0\lambda_{i_0})^{p_{i_0}-1} \prod_{i=\mathtt{l}_{\min}, i\neq i_0}^{\mathtt{l}_{\max}} (n_0\lambda_i)^{p_i} \prod_{j=\mathtt{r}_{\min}}^{\mathtt{r}_{\max}} (n_0\rho_j)^{q_j} \\ & = n_0^{(\sum p_i + \sum q_j)-1} L_{i_0} \lambda_{i_0}^{p_{i_0}-1} \prod_{i=\mathtt{l}_{\min}, i\neq i_0}^{\mathtt{l}_{\max}} \lambda_i^{p_i} \prod_{j=\mathtt{r}_{\min}}^{\mathtt{r}_{\max}} \rho_j^{q_j} \\ & = n_0^{(\sum p_i + \sum q_j)-1} \Pr(\mathring{\mathcal{T}}_t = T) \end{aligned} \tag{28}$$

Substituting (27) and (28) in (26), we get

$$\Pr(\mathring{\mathcal{C}}_t = T) < \frac{\Pr(\mathring{\mathcal{T}}_t = T)}{\left(1 - \frac{\sum p_i}{n_0}\right)^{(\sum p_i)-1} \left(1 - \frac{\sum q_j}{n_0}\right)^{\sum q_j}} \tag{29}$$

The proof is complete once we show that

$$\left(1 - \frac{\sum p_i}{n_0}\right)^{(\sum p_i)-1} \left(1 - \frac{\sum q_j}{n_0}\right)^{\sum q_j} \to 1 \quad \text{as } n \to \infty \tag{30}$$

First, we note that $\sum p_i$ and $\sum q_j$ grow exponentially in t. Hence, there exist constants $\alpha_1, \alpha_2, \beta_1, \beta_2 > 0$ such that

$$\alpha_1 n^{\beta_1} < \sum p_i, \sum q_j < \alpha_2 n^{\beta_2} \tag{31}$$

We have

$$\begin{aligned} 1 > & \left(1 - \frac{\sum p_i}{n_0}\right)^{(\sum p_i)-1} \left(1 - \frac{\sum q_j}{n_0}\right)^{\sum q_j} \\ & > \left(1 - \frac{\alpha_2 n^{\beta_2}}{n_0}\right)^{\alpha_2 n^{\beta_2} - 1} \left(1 - \frac{\alpha_2 n^{\beta_2}}{n_0}\right)^{\alpha_2 n^{\beta_2}} \\ & = \left(1 - \frac{\alpha_2 n^{\beta_2}}{n_0}\right)^{2\alpha_2 n^{\beta_2} - 1} \end{aligned}$$

The proof is complete once we show that

$$\left(1 - \frac{\frac{1}{2}\alpha_2 k \int_0^1 \lambda \mathrm{d}x}{n^{1-\beta_2}}\right)^{n^{\beta_2}} \to 1 \tag{32}$$

For this, we pick the constant $a \in (0, a_{\max}]$ small enough so that $\beta_2 < 0.5$. Observe that for any $\theta > 1$ and $\alpha > 0$, we have

$$\lim_{n\to\infty} \left(1 - \frac{\alpha}{n^\theta}\right)^n = 1 \tag{33}$$

Substituting $m = n^{\beta_2}$ in the left hand side of (32), we have

$$\left(1 - \frac{\frac{1}{2}\alpha_2 k \int_0^1 \lambda \mathrm{d}x}{m^{(1-\beta_2)/\beta_2}}\right)^m$$

which goes to 1 as $m \to \infty$. □

5.7 Strong Secrecy Region

The asymptotic decay of the bit-error probability achieved by the codes in Theorem 5.8 is faster than the inverse cubic decay required for strong secrecy. This directly implies that the duals of the LDPC codes constructed by Algorithm 5.5 achieve strong secrecy on the BEWC under the coset coding scheme.

For a given DDP (λ, ρ), we have constructed a sequence $(\mathcal{C}_n)$ of large-girth LDPC codes based on Ramanujan graphs. For minimum left degree at least three, we showed that for $\varepsilon < \varepsilon_{\text{th}}$, we have

$$\mathbb{E}P_b^{\text{MP}}(\mathcal{C}_n, \varepsilon) = \mathcal{O}(\exp(-\beta n^{a\log(\mathtt{l}_{\min}-1)}))$$

the dual sequence $(\mathcal{C}_n^\perp)$ achieves strong secrecy on BEWC(ξ) for $\xi > 1 - \varepsilon_{\text{th}}$.

5.7.1 Difference Between Regular and Irregular Codes

For *any* large-girth regular LDPC code sequence $(\mathcal{C}_n)$, we have

$$P_b^{\text{MP}}(\mathcal{C}_n, \varepsilon) = \mathcal{O}\left(\exp(-\beta n^{a\log(\mathtt{l}_{\min}-1)})\right)$$

for $\varepsilon < \varepsilon_{\text{th}}$. This means that the dual sequence achieves strong secrecy on BEWC(ξ) for $\xi > 1 - \varepsilon_{\text{th}}$.

For irregular codes *constructed by Algorithm 5.5*, we have shown that

$$\mathbb{E}\left(P_b^{\text{MP}}(\mathcal{C}_n, \varepsilon)\right) = \mathcal{O}\left(\exp(-\beta n^{a\log(\mathtt{1}_{\min}-1)})\right)$$

for $\varepsilon < \varepsilon_{\text{th}}$. Note that our algorithm for irregular LDPC codes incorporates two random permutation functions, thereby producing a random LDPC code. Moreover, there must be at least one code whose bit-error rate is as good as, if not better than, the above average bit-error rate. This means that there must exist codes in the dual sequence that achieve strong secrecy on BEWC(ξ) for $\xi > 1 - \varepsilon_{\text{th}}$.

For the irregular code construction, we also have a concentration around the strong secrecy behavior. For any integer $k > 0$ and a function $f(n) = \Theta(1/n^k)$, we have the following due to Markov's inequality

$$\begin{aligned}\Pr\left(P_b\left(\mathcal{C}_n, \varepsilon\right) \geq f(n)\right) &\leq \frac{\mathbb{E}P_b\left(\mathcal{C}_n, \varepsilon\right)}{f(n)} \\ &\to 0, \quad \text{as } n \to \infty\end{aligned}$$

This means as n increases, the irregular LDPC code output by our algorithm has a bit-error rate $\mathcal{O}(1/n^k)$ with very high probability. The concentration around the strong secrecy behavior follows by setting $k = 3$.

Our result for regular LDPC codes is stronger than that that of irregular LDPC codes. However, irregular LDPC codes are important because they have $1 - \varepsilon_{\text{th}}$ closer to their rate than regular LDPC codes. Therefore, irregular codes are instrumental in achieving strong secrecy at higher rates compared to regular LDPC codes.

5.7.2 Comparison with Other LDPC Code Approaches

Thangaraj, et al. [1] introduced the LDPC code approach to achieve weak secrecy on BEWC(ξ) for $\xi > 1 - \varepsilon_{\text{th}}$. The short-cycle-free LDPC code approach of Chapter 4

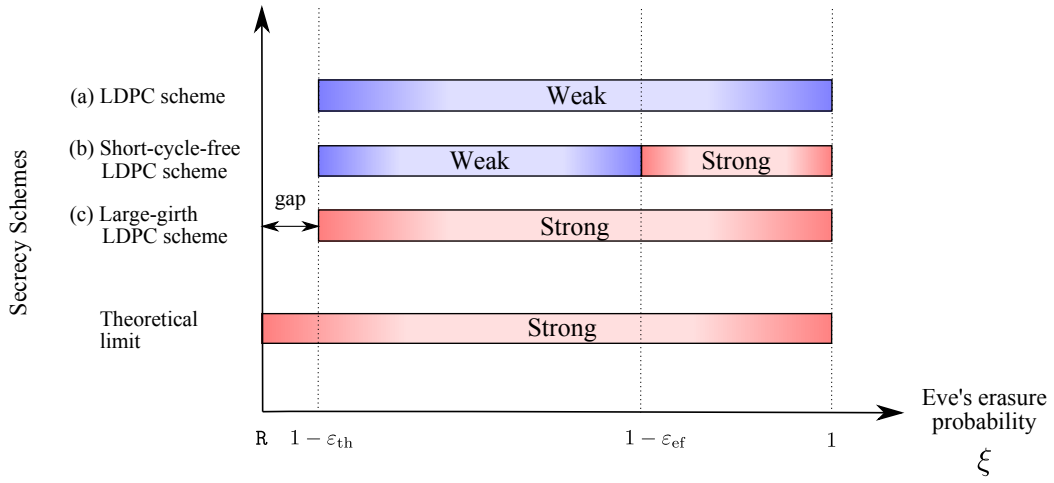


Figure 27: A sketch of the secrecy regions achieved by (a) LDPC codes [1], (b) short-cycle-free LDPC codes, and (c) large-girth LDPC codes.

achieves weak secrecy for $\xi \in (1-\varepsilon_{\text{th}}, 1-\varepsilon_{\text{ef}}]$ and strong secrecy for $\xi \in (1-\varepsilon_{\text{ef}}, 1]$. The large-girth LDPC code approach in this chapter achieves strong secrecy for $\xi \in (1-\varepsilon_{\text{th}}, 1]$, which is an improvement over the other two technique. Note that these secrecy regions are valid only for LDPC code ensembles with $\mathtt{l}_{\min} \geq 3$, i.e., minimum degree at least three. The secrecy regions achieved by these three techniques is plotted in Fig. 27

5.7.3 Gap Between Achievable Region and Secrecy Capacity

For a secret information rate $\mathtt{R}$, we are interested in the minimum value of Eve's erasure probability ξ for which we can ensure strong secrecy over the BEWC using our scheme. Since our proof works only for $\mathtt{l}_{\min} \geq 3$, this involves finding an optimal DDP of rate $\mathtt{R}$ and $\mathtt{l}_{\min} \geq 3$ for which the BEC threshold ε_{th} is as high as possible. It can be noted that $\varepsilon_{\text{th}} < 1-\mathtt{R}$. Most of the capacity achieving DDP sequences require $\mathtt{l}_{\min} = 2$ (e.g., the tornado sequence and the right regular sequence in [61]). Therefore, there is a small gap between the strong secrecy rate achievable by our technique and the secrecy capacity of the BEWC.

For example, when we performed a search on LDPCOPT [42], an online database,

for $R = 0.5$ and $\mathtt{l}_{\min} \geq 3$, we found that an optimal value of $\varepsilon_{\text{th}} = 0.4619$ is achieved by the DDP $\lambda(x) = 0.9043388x^2 + 0.03300419x^{16} + 0.01434268x^{17} + 0.03535427x^{18} + 0.01296008x^{99}$, $\rho(x) = x^{10}$. This means that the duals of the LDPC codes constructed using Algorithm 5.5 will achieve a strong secrecy rate of 0.5 over BEWC(ξ) for all $\xi > 0.5381$. Note that for ξ close to 0.5381, the secrecy capacity of the BEWC is close to 0.5381. Our coding scheme will achieve a secrecy rate of 0.5 over this channel, which is 7% less than the secrecy capacity.

CHAPTER VI

CONCLUSION

Information-theoretic security is the technique of achieving information security by leveraging the randomness in real-world communication channels. This notion of security is relatively new and judging by the number of technical papers published in this topic, information-theoretic security has been attracting growing interest. However, most of the research work in this topic are very theoretical in nature and there are few immediate practical applications of information-theoretic security. Nevertheless, understanding theoretical problems is an important initial step towards practical implementations. In this dissertation, we have studied the problem of designing secure encoders that offer information-theoretic security. This work borrows ideas from the design and analysis of error-correcting codes for noisy channels.

6.1 Contributions

In this dissertation, we deal with the problem of designing secure encoders for a special case of the wiretap channel, namely, the binary erasure wiretap channel. The security criterion in our work, namely, information-theoretic strong secrecy, leads us to a related problem of finding certain good channel codes. It was shown in Chapter 3 that code sequences that achieve $\mathcal{O}\left(\frac{1}{n^2}\right)$ block-error rate or $\mathcal{O}\left(\frac{1}{n^3}\right)$ bit-error rate over the binary erasure channel can be used to design encoders for strong secrecy over the binary erasure wiretap channel.

In Chapter 4, we showed that certain short-cycle-free low-density parity-check codes achieve the required $\mathcal{O}\left(\frac{1}{n^2}\right)$ asymptotic decay in block-error rate. Further, we showed that these codes form an asymptotically significant fraction, which means

that a Monte Carlo algorithm may be used to find these codes.

In Chapter 5, we designed LDPC codes with girth growing logarithmically in block-length. Our explicit construction is based on the large-girth regular graphs invented by Lubotzky, Phillips, and Sarnak [55]. We showed that LDPC codes constructed in this manner achieve a sub-exponential asymptotic bit-error rate of $\mathcal{O}\left(\exp\left(-c_1 n^{c_2}\right)\right)$, with $c_1 > 0$ and $0 < c_2 < 1$, over the binary erasure channel for erasure probabilities below the erasure threshold. These codes satisfy the sufficient condition required for strong secrecy over the binary erasure wiretap channel. We also showed that the strong secrecy region achieved by these codes is a significant improvement over the technique outlined in Chapter 4.

6.2 Future Directions

The following are some of the immediate directions where our work can be extended.

6.2.1 Closing the Gap to Secrecy Capacity

The large-girth LDPC codes designed in this dissertation have minimum variable-node degree at least three. We argued in §5.7.3 that we may be able to use LDPC codes with minimum variable-node degree two to achieve secrecy capacity. The intuition behind this idea is that these a standard ensemble of these LDPC codes has a *bit-error* threshold ε_{th} very close to $1 - \mathtt{R}$, where $\mathtt{R}$ is the code rate—such codes are the so-called "capacity achieving" LDPC codes [61]. However, we use the *block-error* threshold for our strong secrecy result and these codes do not have a block-error threshold for the BEC [39, Thm. 17]. It has been suggested [62] that we may be able to design special ensembles of LDPC codes with minimum variable-node degree two to get a block-error threshold. One of the means of doing this is to use ideas behind *multi-edge-type* LDPC codes [34, §7.1], [63]. One of the setbacks of considering special ensembles is that they have different thresholds compared to the standard ensemble,

which means that finding capacity achieving multi-edge-type LDPC codes is highly non-trivial. Multi-edge-type LDPC codes are still not well-understood and to our knowledge, there are no capacity-achieving multi-edge-type LDPC codes, excepting certain designs [25] that don't fit our strong secrecy requirement. The area of multi-edge-type LDPC codes is a promising field that requires significant research and we believe these codes will play a role in information-theoretic security.

6.2.2 Coding Techniques for Other Wiretap Models

The LDPC code strategy presented in this dissertation achieves strong secrecy only for the binary-erasure wiretap channel. This is because, the security analysis uses a linear algebra approach that is tailored for this channel model. For cases with a noiseless main channel and a binary memoryless wiretap, we proposed an erasure decomposition approach in §3.1.2. However, this approach achieves low secrecy rates. Better secrecy rates are achieved [37] for binary symmetric wiretaps with a more direct approach. In this respect, the polar coding approach [29] is more attractive since it achieves secrecy capacity for all wiretap models with a noiseless main channel.

For wiretap models with a noisy main channel, we are not aware of any current techniques that achieve strong secrecy. One coding approach for these channels is to use an outer channel code to correct errors on the main channel. This approach was used by Rathi, et al. [25] to achieve weak secrecy on a wiretap model with a BEC main channel and an independent BEC wiretap channel, where they modify the LDPC coding approach in [1] to correct errors on the main channel. Achieving strong secrecy on this model by modifying our large-girth LDPC codes in Chapter 5 is a problem worth investigating.

APPENDIX A

LARGE-GIRTH LDPC CODES OVER BMSCS

In Chapter 5, we designed large-girth LDPC codes whose BER decays as $\mathcal{O}\left(-c_1 n^{c_2}\right)$ with growing block length n when they are transmitted over a BEC with the erasure parameter below the threshold. In this chapter, we will show an analogous result when these codes are transmitted over a class of binary-input memoryless symmetric-output channels (BMSCs) when the channel parameter is below the threshold for the given class.

A.1 Fundamentals

In this section, we give a quick introduction of LDPC codes over arbitrary BMSCs. In particular, we provide an overview of the notion of L-densities and the density-evolution analysis of LDPC codes using L-densities. This section is based on the discussion in the *Modern Coding Theory* textbook [34, Ch. 4].

We consider BMSCs whose output alphabet is the extended real number line defined as follows.

Definition A.1 (BMSC)**.** A *binary-input memoryless symmetric-output channel*, denoted by $X \longrightarrow Y$ is a memoryless channel whose input alphabet is $\{-1, 1\}$ and output alphabet is $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$, such that its transition probability, given by the conditional probability density function (pdf) $p_{Y|X}(y|x)$, follows the symmetry requirement

$$p_{Y|X}(y|1) = p_{Y|X}(-y|-1), \qquad \forall y \in \bar{\mathbb{R}}$$

▼

Clearly, any BMSC is uniquely defined by its transition probability $p_{Y|X}(y|x)$.

A.1.1 L-Density Representation of BMSCs

The *log-likelihood ratio (LLR) function* $l(y)$ of a BMSC is defined as

$$l(y) = \ln \frac{p_{Y|X}(y|1)}{p_{Y|X}(y|-1)}$$

Suppose a single random bit X is transmitted over the BMSC, which outputs the random variable Y. Let the LLR random variable L be defined as $L = l(Y)$. It can be shown [34, Lemma 4.7] that L is a sufficient statistic for decoding.

A.1.2 BP Decoding of LDPC Codes

The BP decoder for LDPC codes transmitted over a BMSC works as follows.

1. At iteration $t = 0$, all the variable nodes send the LLR associated with the corresponding channel observation over their incident edges.

2. At iteration $t > 0$, the following two steps take place in this order.

   (a) **Check-node processing:** Along each incident edge, a check node sends the message $\mathcal{L}_{\text{out}}$, which is calculated based on the incoming messages along the other edges, say $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_{k-1}$, as

   $$\tanh\left(\frac{\mathcal{L}_{\text{out}}}{2}\right) = \prod_{i=1}^{k-1} \tanh\left(\frac{\mathcal{L}_i}{2}\right) \tag{34}$$

   (b) **Variable node processing:** Along each incident edge, a variable node sends the LLR message $\mathcal{L}_{\text{out}}$, which is the sum of the incoming LLR messages $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_{j-1}$ along the other edges and the channel output LLR $\mathcal{L}_{\text{channel}}$. That is,

   $$\mathcal{L}_{\text{out}} = \mathcal{L}_{\text{channel}} + \mathcal{L}_1 + \mathcal{L}_2 + \cdots + \mathcal{L}_{j-1} \tag{35}$$

3. At the last iteration, the usual check-node processing takes place, following which a hard-decision estimate of the transmitted bit is calculated at each variable node. At a degree j variable node, the following is calculated.

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{channel}} + \mathcal{L}_1 + \mathcal{L}_2 + \cdots + \mathcal{L}_j$$

The hard-decision output $\hat{X}$ is calculated as follows.

(a) If $\mathcal{L}_{\text{final}} > 0$, $\hat{X} = 1$

(b) If $\mathcal{L}_{\text{final}} < 0$, $\hat{X} = -1$

(c) If $\mathcal{L}_{\text{final}} = 0$, $\hat{X}$ is either -1 or 1 based on the outcome of a fair coin flip.

In this chapter, we will only consider the BP decoder with the flooding schedule (§4.1.3).

A.1.3 Density Evolution Analysis

Due to the symmetry of the BMSC, the BP decoder for LDPC codes can be analyzed by assuming that the all-one codeword (considering the alphabet $\{-1, 1\}$) is transmitted. According to density evolution analysis, the BP decoder performance for LDPC codes closely follows the BP decoder performance for codes associated with *tree ensembles*. Therefore, it is enough to analyze the pdfs of the LLR messages sent along the edges during BP decoding of the tree codes.

L-density

Since the all-one codeword from the tree code was assumed to be transmitted, the corresponding conditional pdfs of all LLR messages are considered. This conditional pdf is given a special name—*L-density*. The L-density associated with the channel observation is denoted by $a_{\text{BMSC}}(z)$. Given the channel observation LLR $\mathcal{L}_{\text{channel}}$, the L-density $a_{\text{BMSC}}(z)$ is the conditional pdf $p_{\mathcal{L}_{\text{channel}}|X}(z|1)$. The L-density $a_{\text{BMSC}}(z)$ is

a channel characteristic that plays an important role in analyzing the performance of codes under iterative decoding.

Considering the BP decoder for the tree code over the given BMSC, let $a_t(z)$ be the L-density of the variable-to-check message at the t^{th} iteration and let $b_t(z)$ be the L-density of the check-to-variable message at the t^{th} iteration.

Check Node Analysis

Given two LLRs $\mathcal{L}_1, \mathcal{L}_2$ with L-densities a_1, a_2, we define the operation $\boxtimes$ as the follows: the result of $a_1 \boxtimes a_2$ is equal to the L-density of the random variable $\mathcal{L}_3$, and it is calculated as:

$$\tanh\left(\frac{\mathcal{L}_3}{2}\right) = \tanh\left(\frac{\mathcal{L}_1}{2}\right)\tanh\left(\frac{\mathcal{L}_2}{2}\right)$$

That is, $a_3 = a_1 \boxtimes a_2$ is the L-density of $\mathcal{L}_3$. It can be easily noted that the $\boxtimes$ operator is both commutative and associative.

Consider a degree-k check node at the check-node processing stage of iteration $t > 0$. The incoming messages are i.i.d. with L-density a_{t-1}. The outgoing message along the k^{th} edge is calculated using (34). Therefore, the L-density of the outgoing message will be

$$b_{\text{out}} = \underbrace{a_{t-1} \boxtimes a_{t-1} \boxtimes \cdots \boxtimes a_{t-1}}_{k-1 \text{ times}} \triangleq a_{t-1}^{\boxtimes(k-1)}$$

Since we are considering the tree-ensemble, the degree of a check node is a random variable distributed according to ρ. Therefore, we can write

$$b_t = \sum_k \rho_k a_{t-1}^{\boxtimes(k-1)} \triangleq \rho(a_{t-1}) \tag{36}$$

Variable Node Analysis

At the 0^{th} iteration of BP decoding, the variable nodes send the channel output LLRs. Therefore, the L-density of the variable-to-check messages at the 0^{th} iteration is equal

to the L-density of the channel. That is,

$$a_0(z) = a_{\text{BMSC}}(z)$$

Consider a degree-j variable node at the variable-node-processing stage of iteration $t > 0$. The incoming messages are i.i.d. with L-densities equal to $b_{t-1}(z)$. Since the processing in (35) involves the addition of LLRs, the L-density of $\mathcal{L}_{\text{out}}$ is given by the convolution of all the associated L-densities. In the case of a degree-j node, it is given by

$$a_{\text{out}} = a_{\text{BMSC}} \circledast \underbrace{b_{t-1} \circledast b_{t-1} \circledast \cdots \circledast b_{t-1}}_{(j-1) \text{ times}} = a_{\text{BMSC}} \circledast b_{t-1}^{\circledast(j-1)}$$

where $a^{\circledast i}$ denotes the convolution of a with itself i times.

In the tree ensemble, the probability distribution of the degree of a non-root variable node is given by λ. Therfore, we have

$$\begin{aligned} a_t &= a_{\text{BMSC}} \circledast \underbrace{\sum_j \lambda_j b_{t-1}^{\circledast(j-1)}}_{\triangleq \lambda(b_{t-1})} \\ &= a_{\text{BMSC}} \circledast \lambda(b_{t-1}) \end{aligned}$$

Using (36), we get the recursive equation

$$a_t = a_{\text{BMSC}} \circledast \lambda(\rho(a_{t-1})) \tag{37}$$

BER Estimate

Consider the root node of $\mathring{\mathcal{T}}_t(\lambda, \rho)$ in the final iteration. The L-density of the final LLR $\mathcal{L}_{\text{final}}$ of the root node is given by

$$c_t = a_{\text{BMSC}} \circledast L(\rho(a_{t-1})) \tag{38}$$

where

$$L\left(b(z)\right) \triangleq \sum_j L_j b^{\circledast j}$$

with L_j being the fraction of degree-j variable nodes in the LDPC code.

The probability of error in the hard-decision decoding is given by the *error probability functional* of the L-density of $\mathcal{L}_{\text{final}}$. The error probability functional of a symmetric L-density a is defined as

$$\begin{aligned}\mathfrak{E}(a) &\triangleq \Pr(L<0)+\frac{1}{2}\Pr(L=0)\\ &= \frac{1}{2}\int_{-\infty}^{\infty} a(z)e^{-|z/2|-z/2}\,\mathrm{d}z\end{aligned}$$

where L is a random variable with L-density a. It can be noted that the probability of error of the root node of $\mathring{\mathcal{T}}_t(\lambda,\rho)$ after t iterations of belief propagation is given by

$$P^{\text{BP}}_{\mathring{\mathcal{T}}_t(\lambda,\rho)}(a_{\text{BMSC}}) = \mathfrak{E}(c_t)$$

By [34, Lemma 4.64], for any L-density a, we have

$$\mathfrak{E}(a) \le \frac{1}{2}\mathfrak{B}(a)$$

where $\mathfrak{B}(a)$ is the *Bhattacharyya functional*, which is given by

$$\mathfrak{B}(a) = \int_{-\infty}^{\infty} a(z)e^{-z/2}\,\mathrm{d}z$$

The Bhattacharyya functional is used in our analysis because it is easier to track when the operations $\circledast$ and $\boxtimes$ are performed than the error functional. We know the following result for the operation $\circledast$.

Lemma A.2 ([34, Lemma 4.63])**.** *Let a and b be two L-densities. Then* $\mathfrak{B}(a \circledast b) = \mathfrak{B}(a)\mathfrak{B}(b)$

For the operation $\boxtimes$, we derive the following result using some of the ideas from Lentmaier, et al. [44].

Lemma A.3 (Based on [44, Appendix I])**.** *Let a and b be two L-densities. Then* $\mathfrak{B}(a \boxtimes b) \le \mathfrak{B}(a) + \mathfrak{B}(b)$

Proof. Let $\mathcal{L}_1, \mathcal{L}_2$ be random variables distributed according to a, b. Note that the correlation between $\mathcal{L}_1$ and $\mathcal{L}_2$ plays no role in the proof. Let $\mathcal{L}_3$ be a random variable such that

$$\tanh\left(\frac{\mathcal{L}_3}{2}\right) = \tanh\left(\frac{\mathcal{L}_1}{2}\right)\tanh\left(\frac{\mathcal{L}_2}{2}\right)$$

Now,

$$\begin{aligned}
\mathfrak{B}(a \circledast b) &= \mathbb{E}\left(e^{-\mathcal{L}_3/2}\right) \\
&= \mathbb{E}\left(\sqrt{e^{-\mathcal{L}_3}}\right) \\
&= \mathbb{E}\left(\sqrt{\frac{1-\tanh\left(\frac{\mathcal{L}_3}{2}\right)}{1+\tanh\left(\frac{\mathcal{L}_3}{2}\right)}}\right) \\
&= \mathbb{E}\left(\sqrt{\frac{1-\tanh\left(\frac{\mathcal{L}_1}{2}\right)\tanh\left(\frac{\mathcal{L}_2}{2}\right)}{1+\tanh\left(\frac{\mathcal{L}_1}{2}\right)\tanh\left(\frac{\mathcal{L}_2}{2}\right)}}\right)
\end{aligned}$$

The proof is complete once we show that

$$\frac{1-\tanh\left(\frac{\mathcal{L}_1}{2}\right)\tanh\left(\frac{\mathcal{L}_2}{2}\right)}{1+\tanh\left(\frac{\mathcal{L}_1}{2}\right)\tanh\left(\frac{\mathcal{L}_2}{2}\right)} \leq \left(e^{-\mathcal{L}_1/2} + e^{-\mathcal{L}_2/2}\right)^2$$

The LHS of the above equation can be written as

$$\begin{aligned}
\frac{1-\tanh\left(\frac{\mathcal{L}_1}{2}\right)\tanh\left(\frac{\mathcal{L}_2}{2}\right)}{1+\tanh\left(\frac{\mathcal{L}_1}{2}\right)\tanh\left(\frac{\mathcal{L}_2}{2}\right)} &= \frac{1-\frac{1-e^{-\mathcal{L}_1}}{1+e^{-\mathcal{L}_1}}\frac{1-e^{-\mathcal{L}_2}}{1+e^{-\mathcal{L}_2}}}{1+\frac{1-e^{-\mathcal{L}_1}}{1+e^{-\mathcal{L}_1}}\frac{1-e^{-\mathcal{L}_2}}{1+e^{-\mathcal{L}_2}}} \\
&= \frac{(1+e^{-\mathcal{L}_1})(1+e^{-\mathcal{L}_2}) - (1-e^{-\mathcal{L}_1})(1-e^{-\mathcal{L}_2})}{(1+e^{-\mathcal{L}_1})(1+e^{-\mathcal{L}_2}) + (1-e^{-\mathcal{L}_1})(1-e^{-\mathcal{L}_2})} \\
&= \frac{e^{-\mathcal{L}_1} + e^{-\mathcal{L}_2}}{1+e^{-(\mathcal{L}_1+\mathcal{L}_2)}} \\
&\leq e^{-\mathcal{L}_1} + e^{-\mathcal{L}_2} \\
&\leq \left(e^{-\mathcal{L}_1/2} + e^{-\mathcal{L}_2/2}\right)^2
\end{aligned}$$

□

A.1.4 Double Exponential Decay of the Estimated Error Probability

Theorem A.4. *For a given DDP (λ, ρ) with minimum left degree $\mathtt{l}_{\min} \geq 3$,*

$$P^{\mathrm{BP}}_{\mathring{\mathcal{T}}_t(\lambda,\rho)}(a_{\mathrm{BMSC}}) = \mathcal{O}\left(\exp(-\beta(\mathtt{l}_{\min}-1)^t)\right) \qquad \text{as } t \to \infty$$

whenever the channel parameter is below the threshold value.

Proof of the theorem

We begin by noting two inequalities. Firstly,

$$\begin{aligned}
\rho(a) &= \sum_k \rho_k a^{\boxplus(k-1)} \\
\Rightarrow \mathfrak{B}(\rho(a)) &= \mathfrak{B}\left(\sum_k \rho_k a^{\boxplus(k-1)}\right) \\
&= \sum_k \rho_k \mathfrak{B}\left(a^{\boxplus(k-1)}\right) \\
&\leq \sum_k \rho_k (k-1)\mathfrak{B}(a) \\
&= (r_{\text{avg}} - 1)\mathfrak{B}(a)
\end{aligned}$$

Secondly,

$$\begin{aligned}
\lambda(a) &= \sum_j \lambda_j a^{\circledast(j-1)} \\
\Rightarrow \mathfrak{B}(\lambda(a)) &= \sum_j \lambda_j \mathfrak{B}\left(a^{\circledast(j-1)}\right) \\
&= \sum_j \lambda_j \left(\mathfrak{B}(a)\right)^{j-1} \\
&\leq \sum_j \lambda_j \left(\mathfrak{B}(a)\right)^{1_{\min}-1} \\
&= \left(\mathfrak{B}(a)\right)^{1_{\min}-1}
\end{aligned}$$

Summarising the above, we have the following two inequalities.

$$\mathfrak{B}(\rho(a)) \leq (r_{\text{avg}} - 1)\mathfrak{B}(a) \tag{39}$$

$$\mathfrak{B}(\lambda(a)) \leq \left(\mathfrak{B}(a)\right)^{1_{\min}-1} \tag{40}$$

Recall that the density evolution recursion is given by

$$\begin{aligned}
a_t &= a_{\text{BMSC}} \circledast \lambda(\rho(a_{t-1})) \\
c_t &= a_{\text{BMSC}} \circledast L(\rho(a_{t-1})) \\
\text{with} \qquad a_0 &= a_{\text{BMSC}}
\end{aligned}$$

We can write

$$\begin{aligned}\mathfrak{B}(a_t) &= \mathfrak{B}(a_{\text{BMSC}})\mathfrak{B}(\lambda(\rho(a_{t-1})))\\ &\overset{(40)}{\leq} \mathfrak{B}(a_{\text{BMSC}})\left(\mathfrak{B}(\rho(a_{t-1}))\right)^{\mathbb{1}_{\min}-1}\\ &\overset{(39)}{\leq} \mathfrak{B}(a_{\text{BMSC}})\left((r_{\text{avg}}-1)\mathfrak{B}(a_{t-1})\right)^{\mathbb{1}_{\min}-1}\\ \Rightarrow \mathfrak{B}(a_t) &\leq A(\mathfrak{B}(a_{t-1}))^{\mathbb{1}_{\min}-1} \qquad (41)\end{aligned}$$

where $A \triangleq \mathfrak{B}(a_{\text{BMSC}})(r_{\text{avg}}-1)^{\mathbb{1}_{\min}-1}$ is a positive quantity that depends only on the channel and the DDP (λ, ρ).

Since the channel parameter of the BMSC is "below" the threshold, we have $\mathfrak{E}(a_t) \to 0$ as $l \to \infty$. This also means that $\mathfrak{B}(a_t) \to 0$ as $l \to \infty$ because

$$\mathfrak{E}(a) \leq \frac{1}{2}\mathfrak{B}(a) \leq \sqrt{\mathfrak{E}(a)(1-\mathfrak{E}(a))}$$

for any L-density a.

Since $\mathfrak{B}(a_t) \to 0$, there exists an $R \in \mathbb{N}$ such that

$$z_R \triangleq \mathfrak{B}(a_R) < \min\left\{\frac{1}{r_{\text{avg}}-1}, \frac{1}{A^{1/(\mathbb{1}_{\min}-2)}}\right\}$$

For $i > 0$, let

$$z_{R+i} = A z_{R+i-1}^{\mathbb{1}_{\min}-1}$$

Claim A.5. *For $i \geq 0$, we have $\mathfrak{B}(a_{R+i}) \leq z_{R+i}$.*

Proof. We prove this by induction. The base case is for $i = 0$ and it is true by our choice of R.

Assume that the claim is true for some $i \geq 0$. We have

$$\begin{aligned}\mathfrak{B}(a_{R+i+1}) &\leq \mathfrak{B}(a_{\text{BMSC}})\left((r_{\text{avg}}-1)\mathfrak{B}(a_{R+i})\right)^{\mathbb{1}_{\min}-1}\\ &\leq \mathfrak{B}(a_{\text{BMSC}})\left((r_{\text{avg}}-1)z_{R+i}\right)^{\mathbb{1}_{\min}-1}\\ &= z_{R+i+1}\end{aligned}$$

This proves the claim. □

Continuing with the main proof, we have

$$\begin{aligned}
z_{R+i} &= A^{1+(\mathtt{l}_{\min}-1)+(\mathtt{l}_{\min}-1)^2+\cdots+(\mathtt{l}_{\min}-1)^{i-1}} z_R^{(\mathtt{l}_{\min}-1)^i} \\
&= A^{\frac{(\mathtt{l}_{\min}-1)^i-1}{\mathtt{l}_{\min}-2}} z_R^{(\mathtt{l}_{\min}-1)^i} \\
&= A^{\frac{-1}{\mathtt{l}_{\min}-2}} \left(A^{\frac{1}{\mathtt{l}_{\min}-2}} x_R\right)^{(\mathtt{l}_{\min}-1)^i} \\
&= A^{\frac{-1}{\mathtt{l}_{\min}-2}} \exp\left((\mathtt{l}_{\min}-1)^i \left(\frac{\log A}{\mathtt{l}_{\min}-2} + \log x_R\right)\right) \\
&= A^{\frac{-1}{\mathtt{l}_{\min}-2}} \exp\left(-\alpha_R (\mathtt{l}_{\min}-1)^i\right)
\end{aligned}$$

Due to our choice of R, $\alpha_R \triangleq \frac{-1}{\mathtt{l}_{\min}-2} \log A - \log z_R$ is positive. For $t \geq R$, we have

$$\begin{aligned}
z_t &= A^{\frac{-1}{\mathtt{l}_{\min}-2}} \exp\left(-\alpha_R (\mathtt{l}_{\min}-1)^{t-R}\right) \\
&= A^{\frac{-1}{\mathtt{l}_{\min}-2}} \exp\left(-\frac{\alpha_R}{(\mathtt{l}_{\min}-1)^R} (\mathtt{l}_{\min}-1)^t\right) \\
&= A^{\frac{-1}{\mathtt{l}_{\min}-2}} \exp\left(-\beta (\mathtt{l}_{\min}-1)^t\right)
\end{aligned}$$

Note that $\beta \triangleq \frac{\alpha_R}{(\mathtt{l}_{\min}-1)^R} > 0$. Therefore, we have

$$\mathfrak{B}(a_t) = \mathcal{O}\left(\exp(-\beta(\mathtt{l}_{\min}-1)^t)\right) \quad \text{as } t \to \infty$$

It is easy to see that the following is true.

$$\mathfrak{B}(c_t) \leq A(\mathfrak{B}(a_{t-1}))^{\mathtt{l}_{\min}-1}$$

This inequality is analogous to (41) and therefore we can repeat the above proof to show that

$$\mathfrak{B}(c_t) = \mathcal{O}\left(\exp(-\beta(\mathtt{l}_{\min}-1)^t)\right) \quad \text{as } t \to \infty$$

Theorem A.6. *For a given DDP (λ, ρ) with minimum left degree $\mathtt{l}_{\min} \geq 3$, the sequence of large-girth (λ, ρ)-irregular LDPC codes $(\mathcal{C}_n)$ created using Algorithm 5.5, when transmitted over an arbitrary BMSC with its L-density a_{BMSC} below the threshold in the channel class, achieve an expected probability of bit-error that decays as*

$$\mathbb{E}P_b^{\mathrm{MP}}(\mathcal{C}_n, a_{\mathrm{BMSC}}) = \mathcal{O}\left(\exp(-c_1 n^{c_2})\right) \tag{42}$$

for some positive constants c_1, c_2.

Proof. Given Theorem A.4, the proof is identical to the proof of Theorem 5.8. □

REFERENCES

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

[2] E. Foundation, M. Loukides, and J. Gilmore, *Cracking DES: Secrets of encryption research, wiretap politics and chip design.* O'Reilly & Associates, Inc. Sebastopol, CA, USA, 1998.

[3] T. Cover and J. Thomas, *Elements of information theory.* New York: Wiley, 1991.

[4] C. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, pp. 656–715, Oct. 1949.

[5] G. S. Vernam, "Cipher printing telegraph systems for secret wire and radio telegraphic communications," *American Institute of Electrical Engineers, Transactions of the*, vol. XLV, pp. 295–301, Jan. 1926.

[6] A. D. Wyner, "The wire-tap channel," *Bell System Technical Journal*, vol. 54, no. 8, pp. 1355–1367, Oct. 1975.

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

[8] J. Körner and K. Marton, "Comparison of two noisy channels," in *Proc. of Topics in Information Theory*, Keszthely, Hungary, 1977, pp. 411–423.

[9] I. Csiszár, "Almost independence and secrecy capacity," *Problems of Information Transmission*, vol. 32, no. 1, pp. 40–47, Jan.–Mar. 1996.

[10] U. M. Maurer and S. Wolf, "Information-theoretic key agreement: From weak to strong secrecy for free," in *Advances in Cryptology - Eurocrypt 2000*, Lecture Notes in Computer Science. B. Preneel, 2000, p. 351.

[11] L. Trevisan, "Construction of extractors using pseudo-random generators (extended abstract)," in *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, ser. STOC '99. New York, NY, USA: ACM, 1999, pp. 141–148. [Online]. Available: http://doi.acm.org.www.library.gatech.edu:2048/10.1145/301250.301289

[12] S. Vadhan, "Extracting all the randomness from a weakly random source," Massachusetts Institute of Technology, Cambridge, MA, USA, Tech. Rep., 1998. [Online]. Available: citeseer.ist.psu.edu/530085.html

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

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

[15] G. Blakley, "Safeguarding cryptographic keys," *AFIPS National Computer Conference*, pp. 313–317, 1979.

[16] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[17] R. J. McEliece and D. V. Sarwate, "On sharing secrets and reed-solomon codes," *Commun. ACM*, vol. 24, no. 9, pp. 583–584, 1981.

[18] E. Karnin, J. Greene, and M. Hellman, "On secret sharing systems," *IEEE Transactions on Information Theory*, vol. 29, no. 1, pp. 35–41, Jan. 1983.

[19] R. Singleton, "Maximum distance q -nary codes," *Information Theory, IEEE Transactions on*, vol. 10, no. 2, pp. 116–118, Apr. 1964.

[20] E. F. Brickell and D. M. Davenport, "On the classification of ideal secret sharing schemes," *J. Cryptology*, vol. 4, no. 2, pp. 123–134, 1991.

[21] K. Kurosawa, K. Okada, K. Sakano, W. Ogata, and S. Tsujii, "Nonperfect secret sharing schemes and matroids," in *Advances in Cryptology - AUSCRYPT '92*, 1994.

[22] J. G. Oxley, *Matroid Theory (Oxford Graduate Texts in Mathematics)*. Oxford University Press, USA, 2006.

[23] J. L. Massey, "Minimal codewords and secret sharing," in *Proceedings of the 6th Joint Swedish-Russian International Workshop on Information Theory*, 1993, pp. 276–279.

[24] ——, "Some applications of coding theory in cryptography," in *Codes and Ciphers: Cryptography and Coding IV*, 1995, pp. 33–47.

[25] V. Rathi, M. Andersson, R. Thobaben, J. Kliewer, and M. Skoglund, "Two edge type LDPC codes for the wiretap channel," in *Signals, Systems and Computers, 2009 Conference Record of the Forty-Third Asilomar Conference on*, Nov. 2009, pp. 834–838.

[26] R. Liu, Y. Liang, H. V. Poor, and P. Spasojević, "Secure nested codes for type II wiretap channels," in *Proceedings of IEEE Information Theory Workshop*, Lake Tahoe, California, USA, Sep. 2007, pp. 337–342.

[27] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *Information Theory, IEEE Transactions on*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[28] H. Mahdavifar and A. Vardy, "Achieving the secrecy capacity of wiretap channels using polar codes," in *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, Jun. 2010, pp. 913–917.

[29] ——, "Achieving the secrecy capacity of wiretap channels using polar codes," *IEEE Trans. Inf. Theory*, 2011, to appear. [Online]. Available: http://arxiv.org/abs/1001.0210v2

[30] M. Andersson, V. Rathi, R. Thobaben, J. Kliewer, and M. Skoglund, "Nested polar codes for wiretap and relay channels," *Communications Letters, IEEE*, vol. 14, no. 8, pp. 752–754, Aug. 2010.

[31] E. Hof and S. Shamai, "Secrecy-achieving polar-coding," in *Information Theory Workshop (ITW), 2010 IEEE*, Sep. 2010, pp. 1–5. [Online]. Available: http://arxiv.org/abs/1005.2759

[32] O. Koyluoglu and H. El Gamal, "Polar coding for secure transmission and key agreement," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2010 IEEE 21st International Symposium on*, Sep. 2010, pp. 2698–2703.

[33] C. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, Jul. 1948.

[34] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2008.

[35] M. Bloch and J. Barros, *Physical-Layer Security: From Information Theory to Security Engineering*. Cambridge University Press, 2011, to be published.

[36] A. T. Suresh, A. Subramanian, A. Thangaraj, M. Bloch, and S. W. McLaughlin, "Strong secrecy for erasure wiretap channels," in *Information Theory Workshop, 2010. ITW 2010. IEEE*, Aug. 2010. [Online]. Available: http://arxiv.org/abs/1004.5540

[37] A. Subramanian, A. T. Suresh, S. Raj, A. Thangaraj, M. Bloch, and S. W. McLaughlin, "Strong and weak secrecy in wiretap channels," in *Turbo Codes and Iterative Information Processing, 2010 6th International Symposium on*, 2010, to appear.

[38] R. G. Gallager, *Low-Density Parity-Check Codes*. The MIT Press, 1963.

[39] A. Orlitsky, K. Viswanathan, and J. Zhang, "Stopping set distribution of LDPC code ensembles," *IEEE Transactions on Information Theory*, vol. 51, no. 3, pp. 929–953, Mar. 2005.

[40] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

[41] B. D. McKay, N. C. Wormald, and B. Wysocka, "Short cycles in random regular graphs," *Electr. J. Comb.*, vol. 11, no. 1, 2004.

[42] R. Urbanke and A. Amraoui. (2010, Jun.) LDPCOPT - a fast and accurate degree distribution optimizer for LDPC code ensembles. Online database. [Online]. Available: http://ipgdemos.epfl.ch/ldpcopt/

[43] A. Subramanian, A. Thangaraj, M. Bloch, and S. McLaughlin, "Strong secrecy on the binary erasure wiretap channel using large-girth LDPC codes," *Information Forensics and Security, IEEE Transactions on*, 2011, to appear.

[44] M. Lentmaier, D. Truhachev, K. Zigangirov, and D. Costello, "An analysis of the block error probability performance of iterative decoding," *Information Theory, IEEE Transactions on*, vol. 51, no. 11, pp. 3834–3855, Nov. 2005.

[45] S. B. Korada and R. L. Urbanke, "Exchange of limits: Why iterative decoding works," *Information Theory, IEEE Transactions on*, vol. 57, no. 4, pp. 2169–2187, Apr. 2011.

[46] X.-Y. Hu, E. Eleftheriou, and D. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *Information Theory, IEEE Transactions on*, vol. 51, no. 1, pp. 386–398, Jan. 2005.

[47] L. S. Chandran, "A high girth graph construction," *SIAM J. Discret. Math.*, vol. 16, no. 3, pp. 366–370, 2003.

[48] K. M. Krishnan, R. Singh, L. S. Chandran, and P. Shankar, "A combinatorial family of near regular LDPC codes," in *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, Jun. 2007, pp. 761–765.

[49] F. Lazebnik and V. A. Ustimenko, "Explicit construction of graphs with an arbitrary large girth and of large size," *Discrete Appl. Math.*, vol. 60, no. 1-3, pp. 275–284, 1995.

[50] F. Lazebnik, V. Ustimenko, and A. Woldar, "A new series of dense graphs of high girth," *American Mathematical Society*, vol. 32, no. 1, 1995.

[51] J.-L. Kim, U. Peled, I. Perepelitsa, V. Pless, and S. Friedland, "Explicit construction of families of LDPC codes with no 4-cycles," *Information Theory, IEEE Transactions on*, vol. 50, no. 10, pp. 2378–2388, Oct. 2004.

[52] G. Margulis, "Explicit constructions of graphs without short cycles and low density codes," *Combinatorica*, vol. 2, no. 1, pp. 71–78, 1982.

[53] J. Rosenthal and P. Vontobel, "Constructions of regular and irregular LDPC codes using Ramanujan graphs and ideas from Margulis," in *Information Theory, 2001. Proceedings. 2001 IEEE International Symposium on*, 2001, p. 4.

[54] J. Rosenthal and P. O. Vontobel, "Constructions of LDPC codes using Ramanujan graphs and ideas from Margulis," in *in Proc. of the 38-th Allerton Conference on Communication, Control, and Computing*, 2000, pp. 248–257.

[55] A. Lubotzky, R. Phillips, and P. Sarnak, "Ramanujan graphs," *Combinatorica*, vol. 8, no. 3, pp. 261–277, 1988.

[56] G. Davidoff, P. Sarnak, and A. Valette, *Elementary Number Theory, Group Theory and Ramanujan Graphs (London Mathematical Society Student Texts)*. Cambridge University Press, 2003.

[57] Y. Glasner, "Ramanujan graphs with small girth," *Combinatorica*, vol. 23, pp. 487–502, 2003, 10.1007/s00493-003-0029-9. [Online]. Available: http://dx.doi.org/10.1007/s00493-003-0029-9

[58] M. Sipser and D. Spielman, "Expander codes," *Information Theory, IEEE Transactions on*, vol. 42, no. 6, pp. 1710–1722, Nov. 1996.

[59] B. Bollobás, *Extremal Graph Theory*. Dover Publications, 2004.

[60] T. M. Apostol, *Introduction to Analytic Number Theory (Undergraduate Texts in Mathematics)*. Springer, 1976.

[61] P. Oswald and A. Shokrollahi, "Capacity-achieving sequences for the erasure channel," *Information Theory, IEEE Transactions on*, vol. 48, no. 12, pp. 3017–3028, Dec. 2002.

[62] H. Jin and T. Richardson, "Block error iterative decoding capacity for LDPC codes," in *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*, Sep. 2005, pp. 52–56.

[63] T. Richardson and R. Urbanke, "Multi-edge type LDPC codes," April 2004, unpublished. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.106.7310

VITA

Arunkumar Subramanian received his Bachelor of Technology (B.Tech) degree in Electrical Engineering from the Indian Institute of Technology, Madras, India in May 2005. In August 2005, he joined the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, USA as a graduate student. In May 2008, he received his Master of Science (M.S.) degree and in 2011, he received his Doctor of Philosophy (Ph.D.), both in Electrical and Computer Engineering from the Georgia Institute of Technology.